OIPE
NOV 0 5 2004
PATENT & TRADEMARK OFFICE

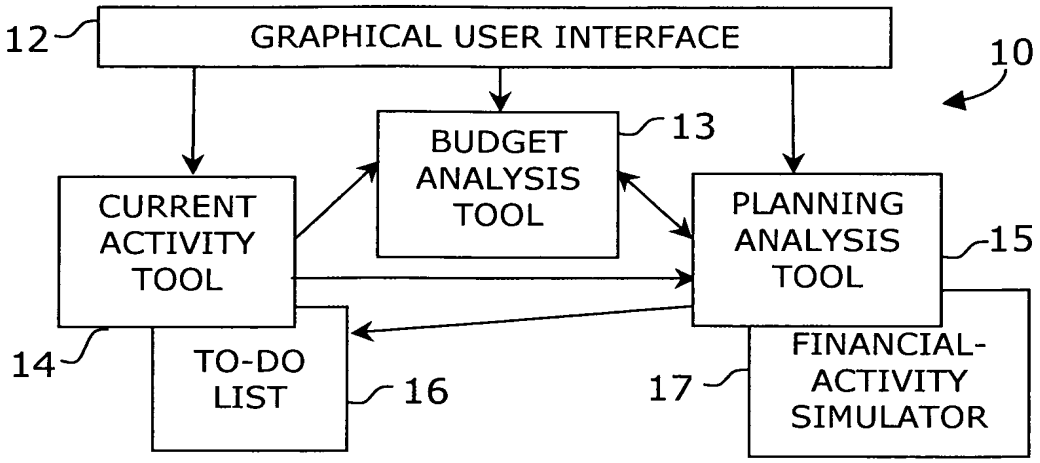

FIG. 1

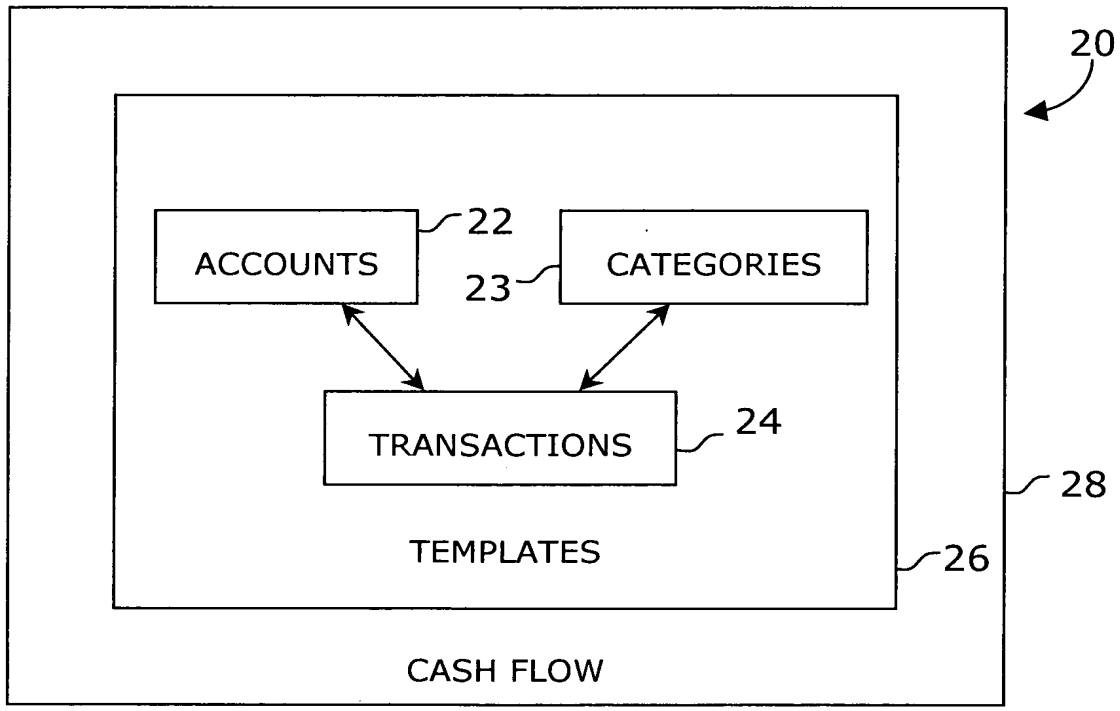

FIG. 2

OIPE JC156
NOV 0 5 2004
PATENT & TRADEMARK OFFICE

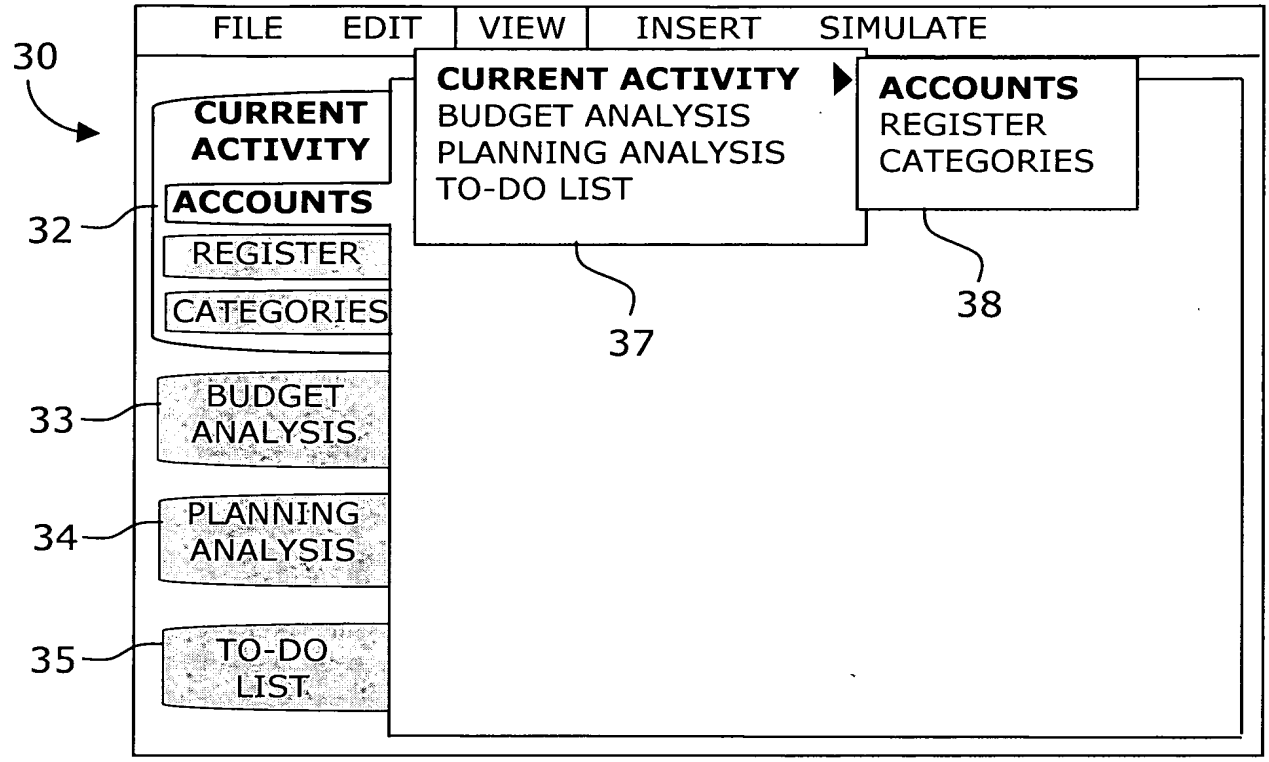

FIG. 3

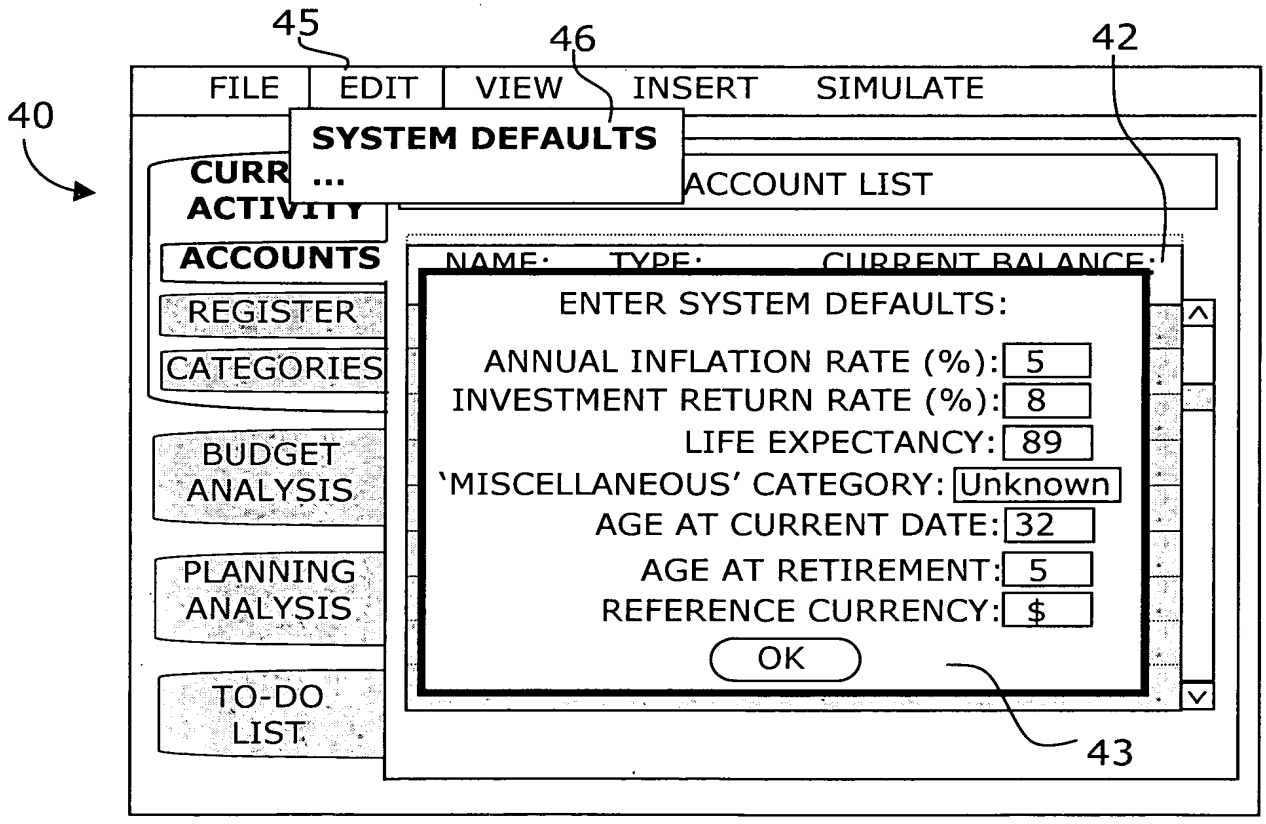

FIG. 4

OIPE JC156 NOV 05 2004 PATENT & TRADEMARK OFFICE

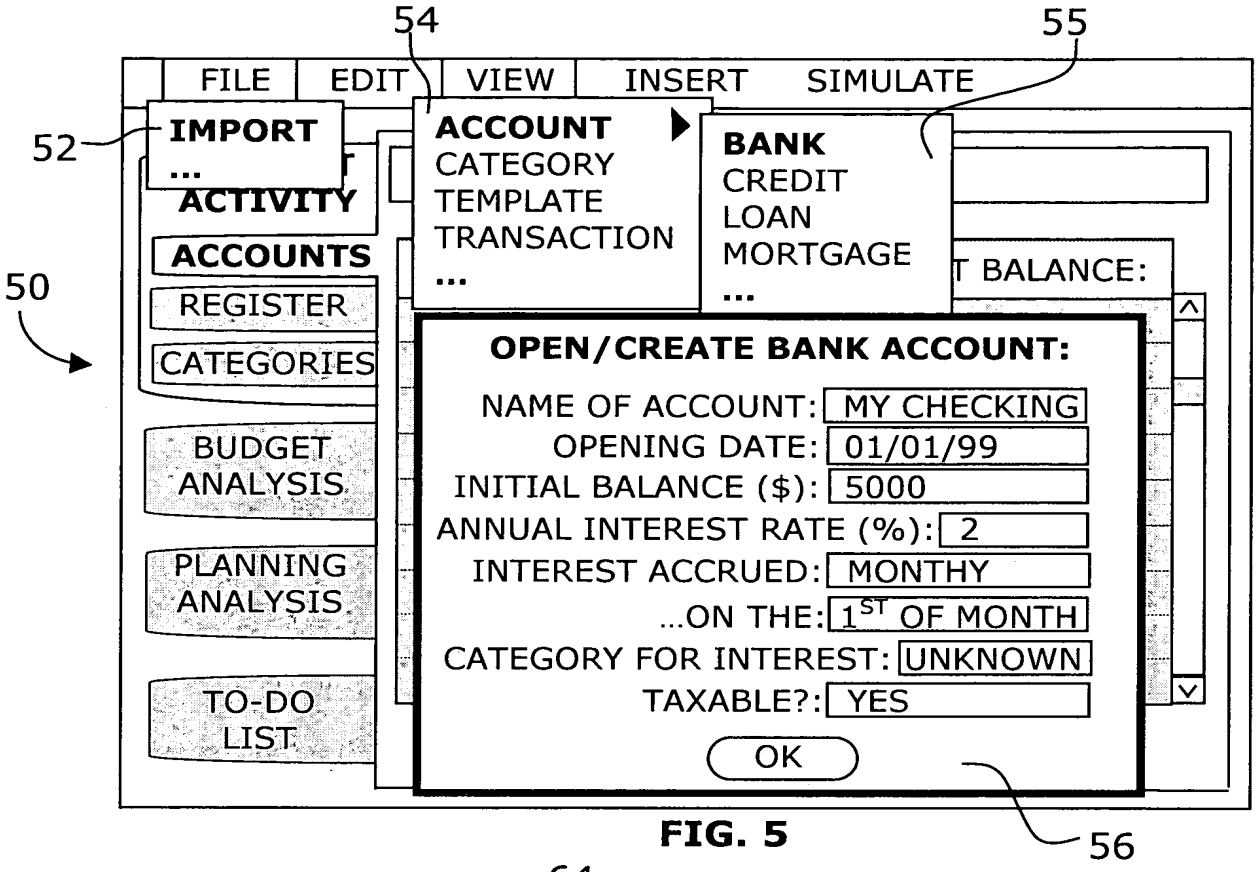

FIG. 5

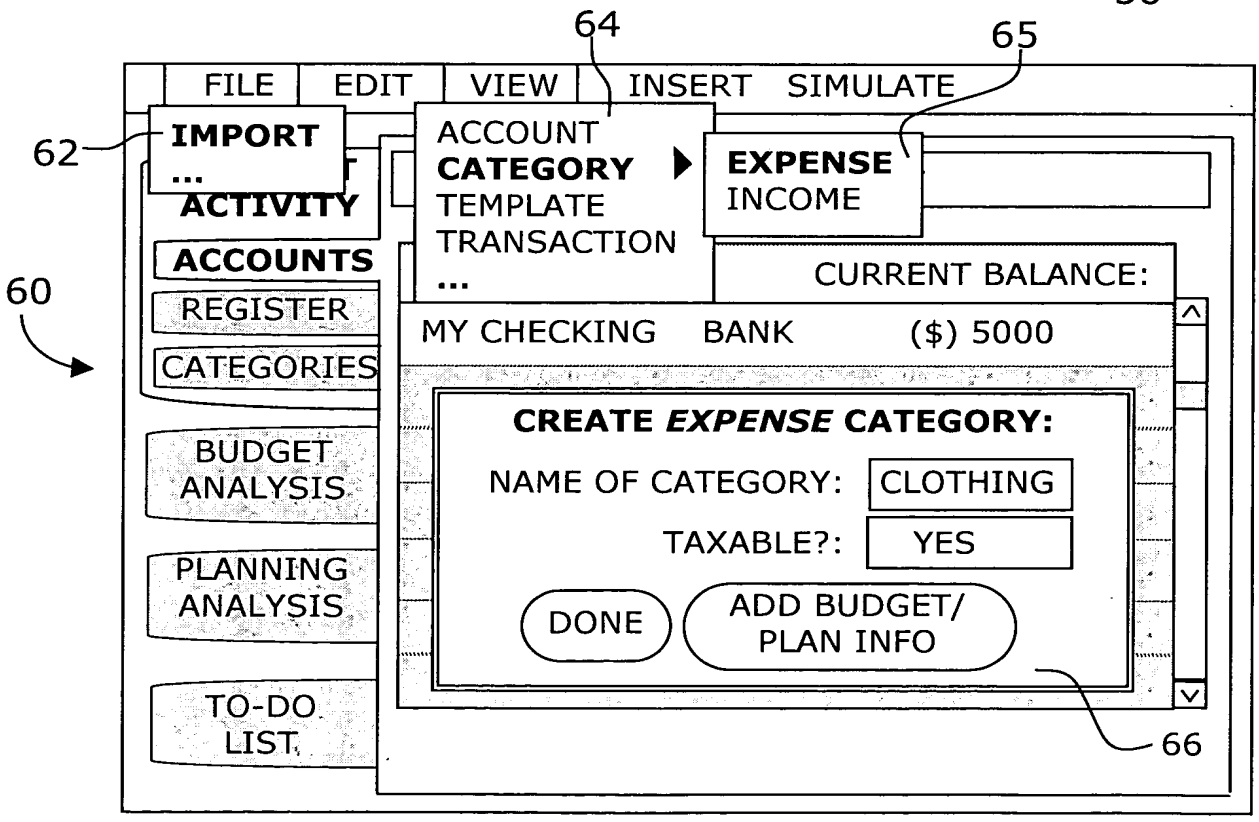

FIG. 6

OIPE JC156
NOV 05 2004
PATENT & TRADEMARK OFFICE

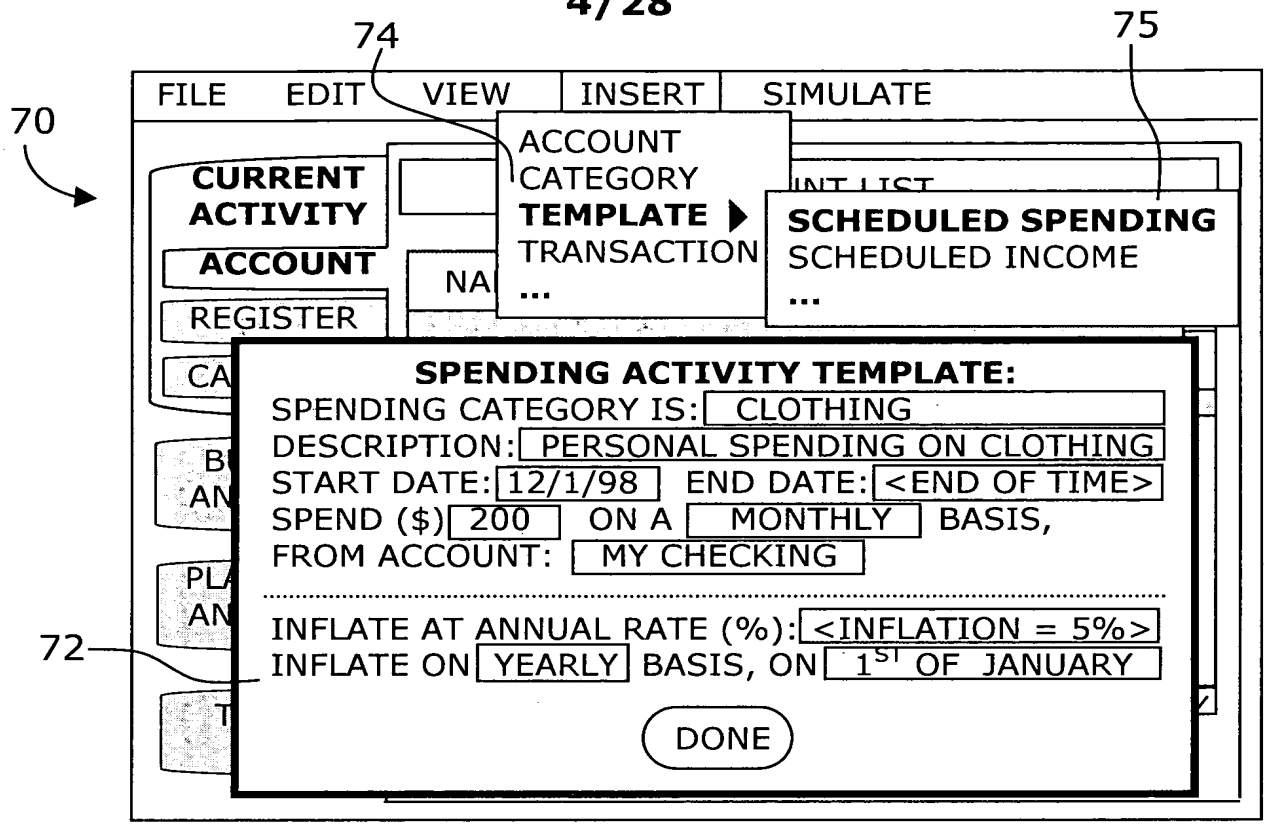

FIG. 7

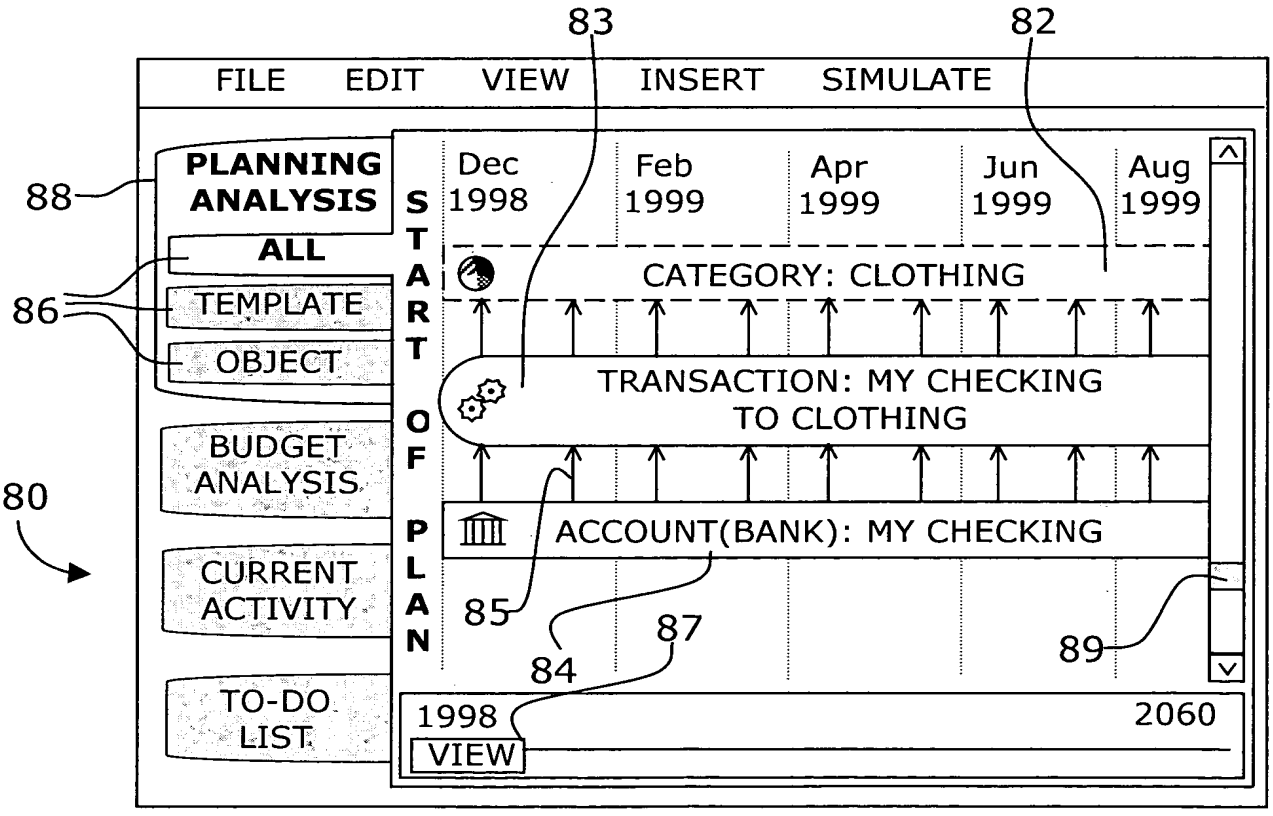

FIG. 8

OIPE JC156
NOV 05 2004
PATENT & TRADEMARK OFFICE

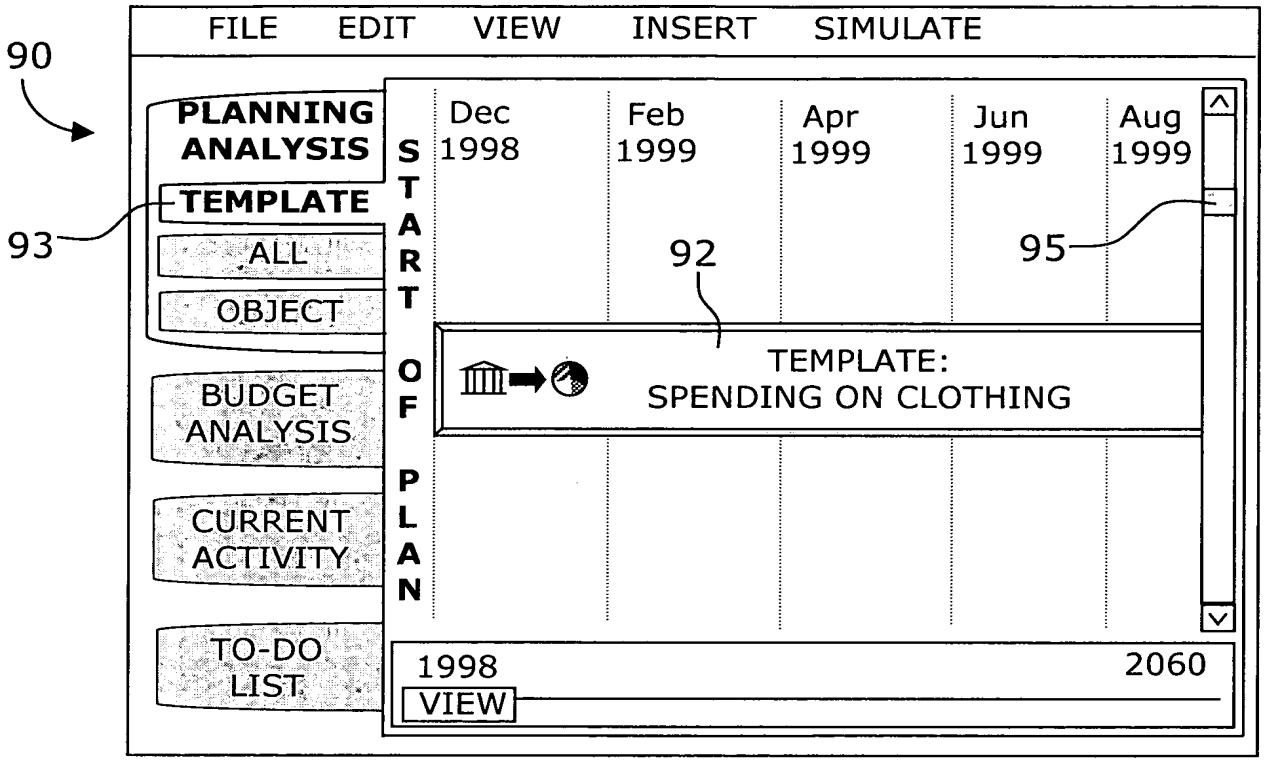

FIG. 9

PATENT & TRADEMARK OFFICE
NOV 05 2004
OIPE
JC166

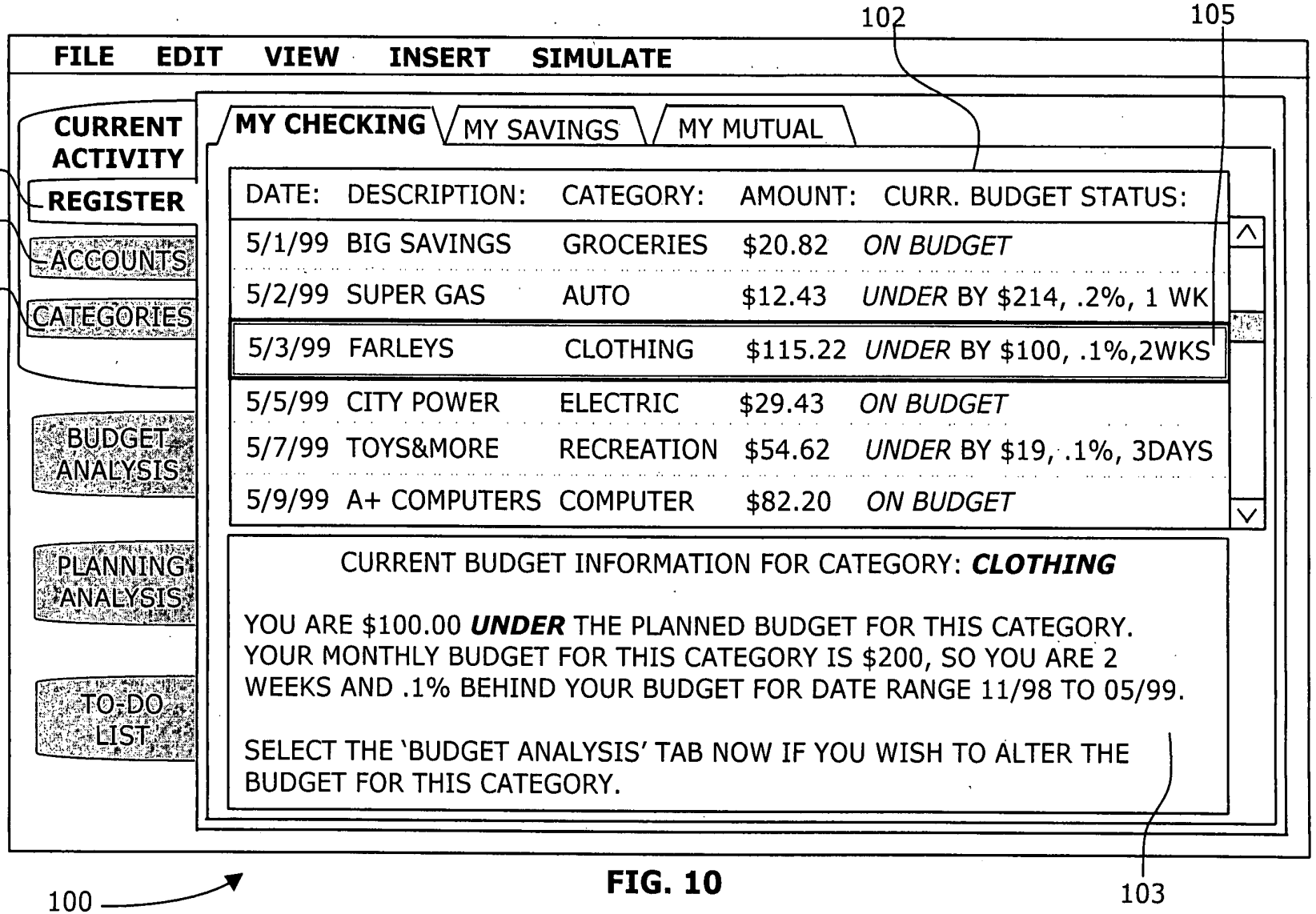

FIG. 10

OIPE JC156 NOV 05 2004 PATENT & TRADEMARK OFFICE

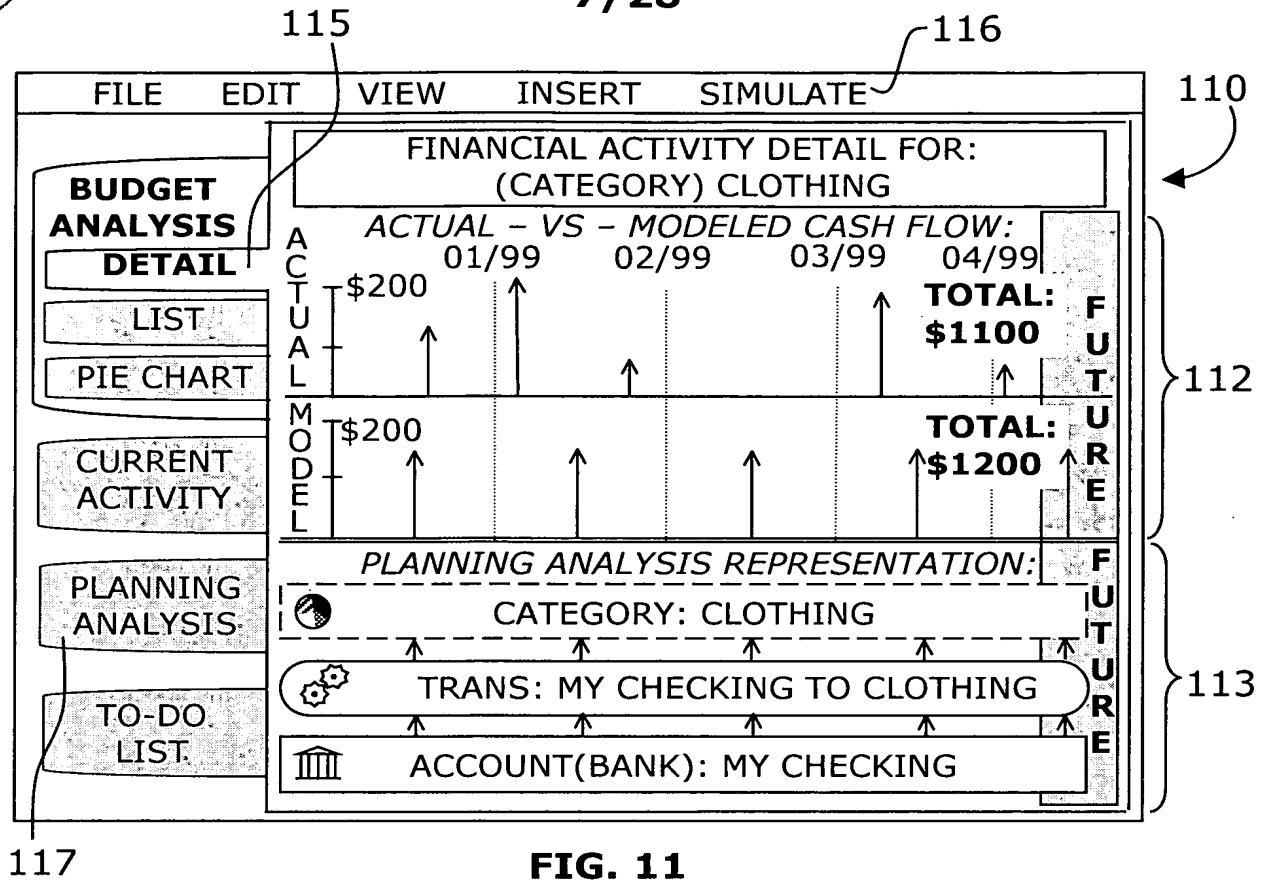

FIG. 11

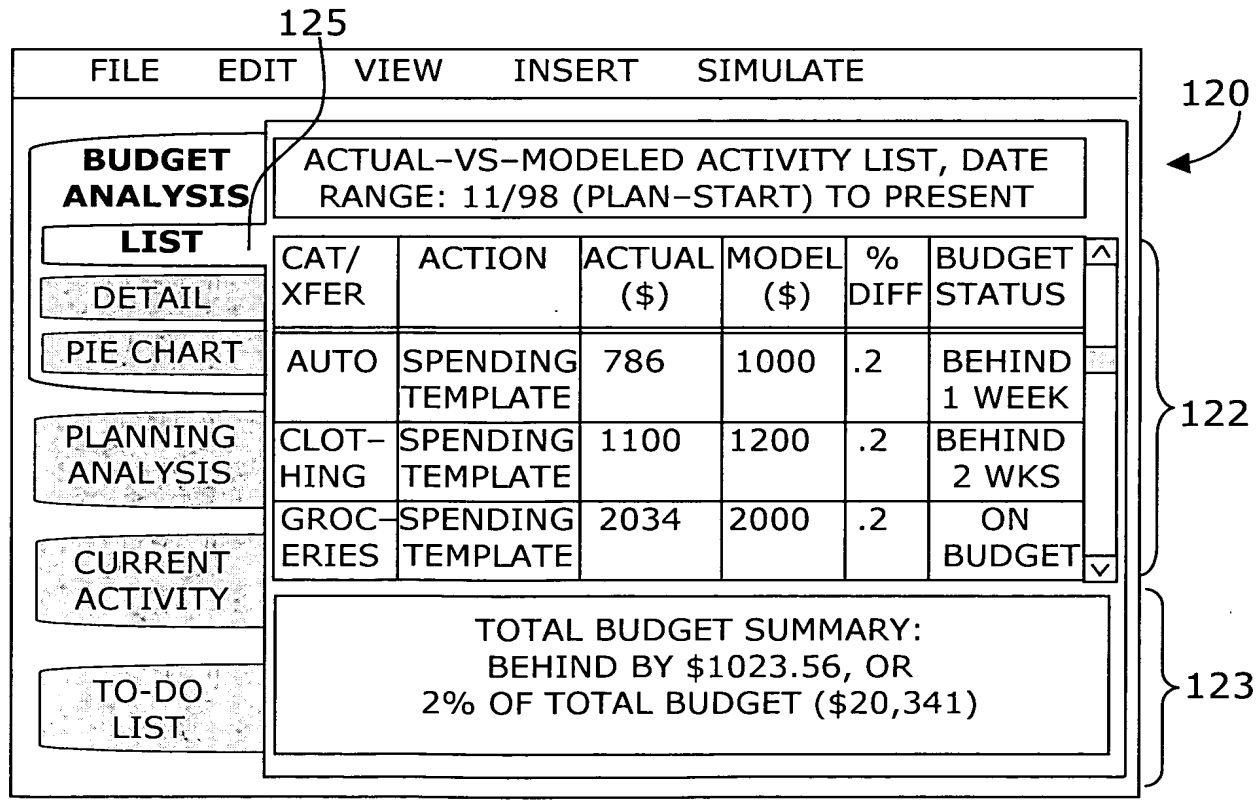

FIG. 12

OIPE JC156 NOV 05 2004 PATENT & TRADEMARK OFFICE

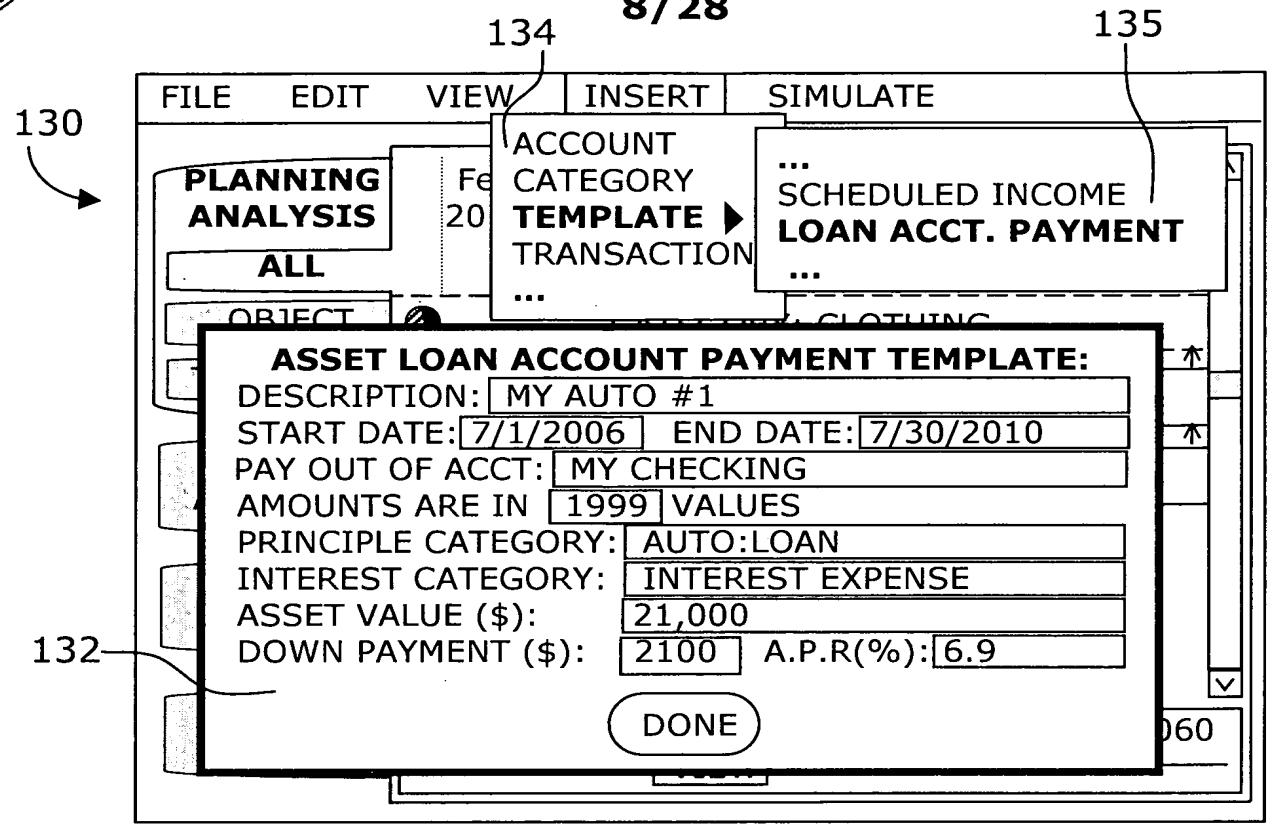

FIG. 13

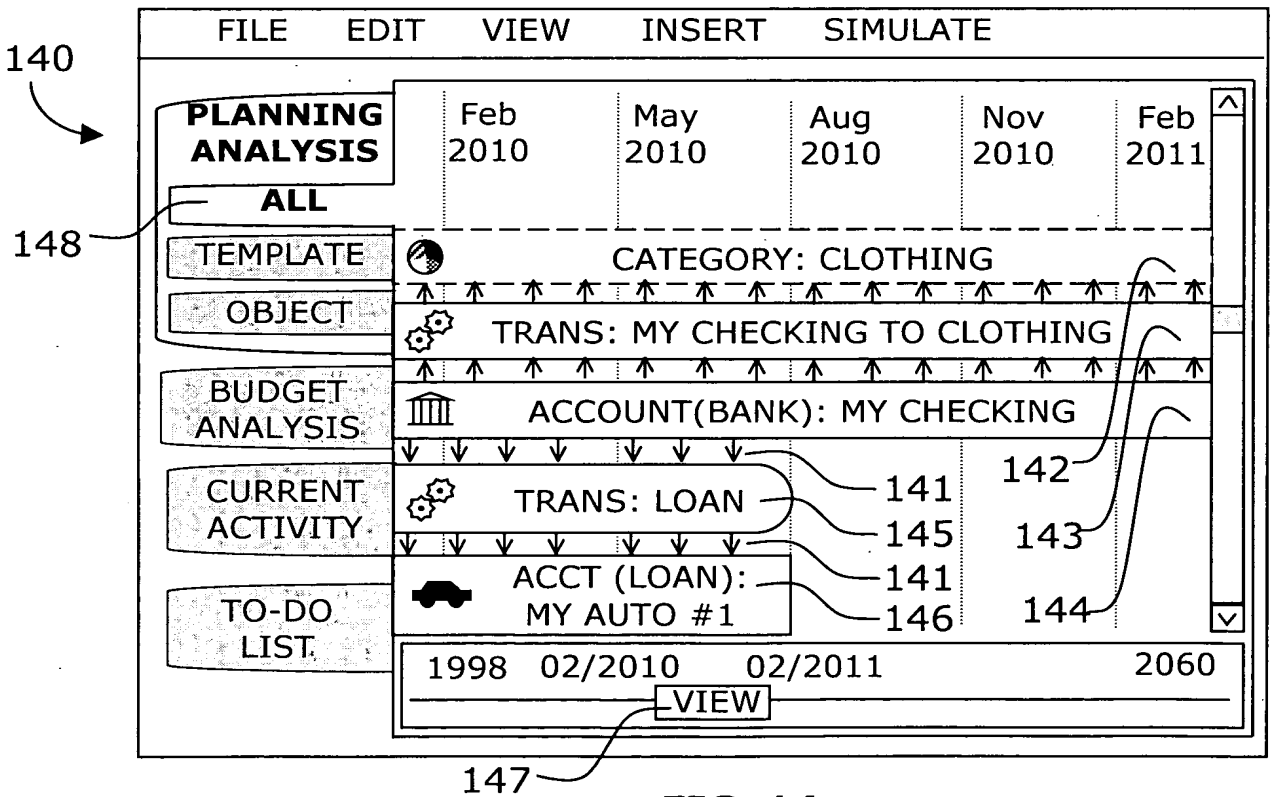

FIG. 14

OIPE JC156
NOV 05 2004
PATENT & TRADEMARK OFFICE

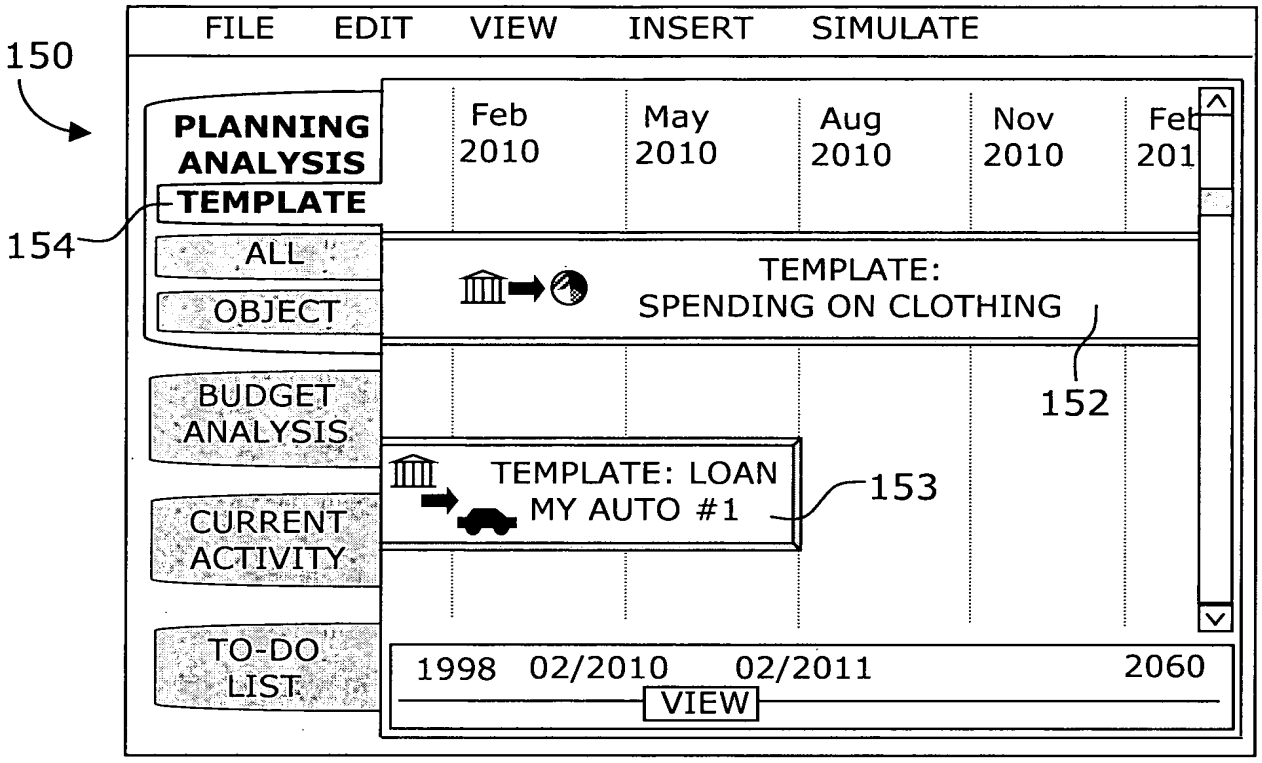

FIG. 15

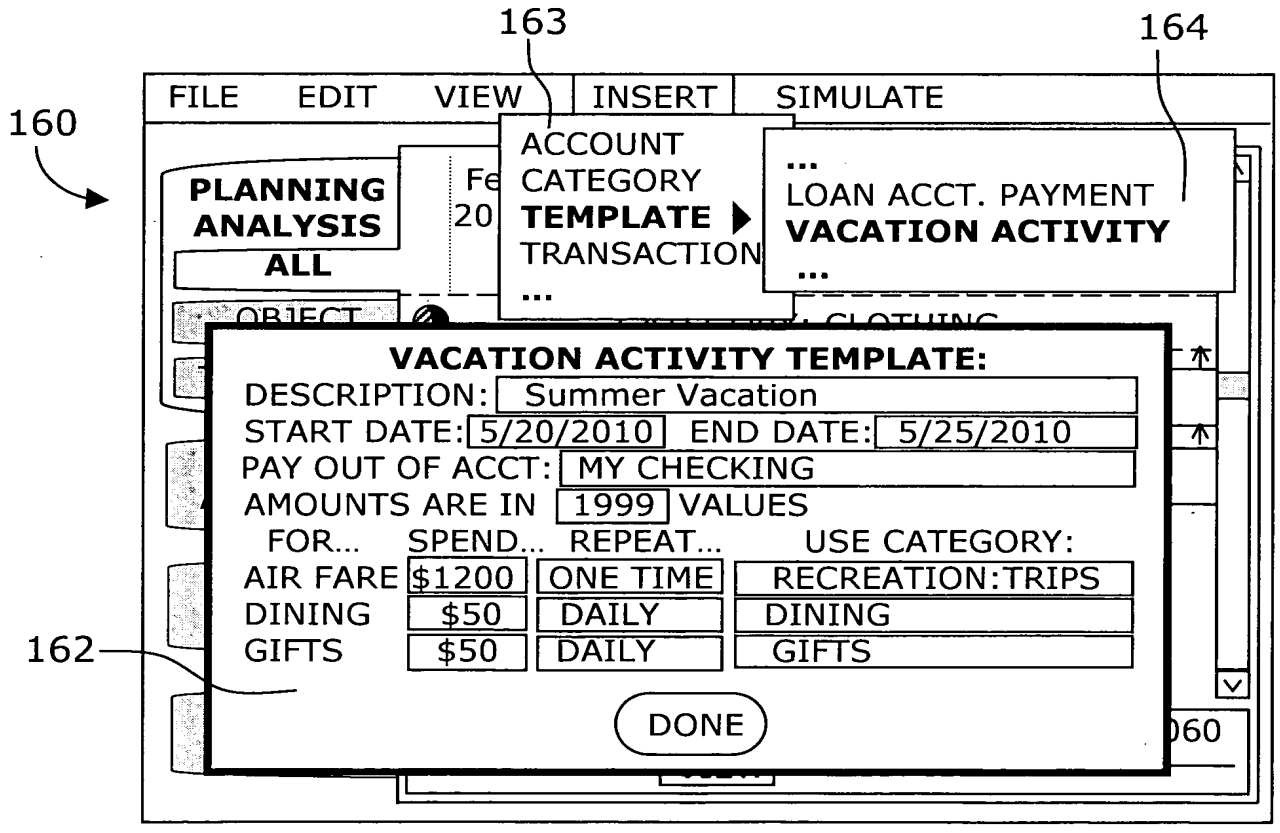

FIG. 16

OIPE JC166 NOV 05 2004 PATENT & TRADEMARK OFFICE

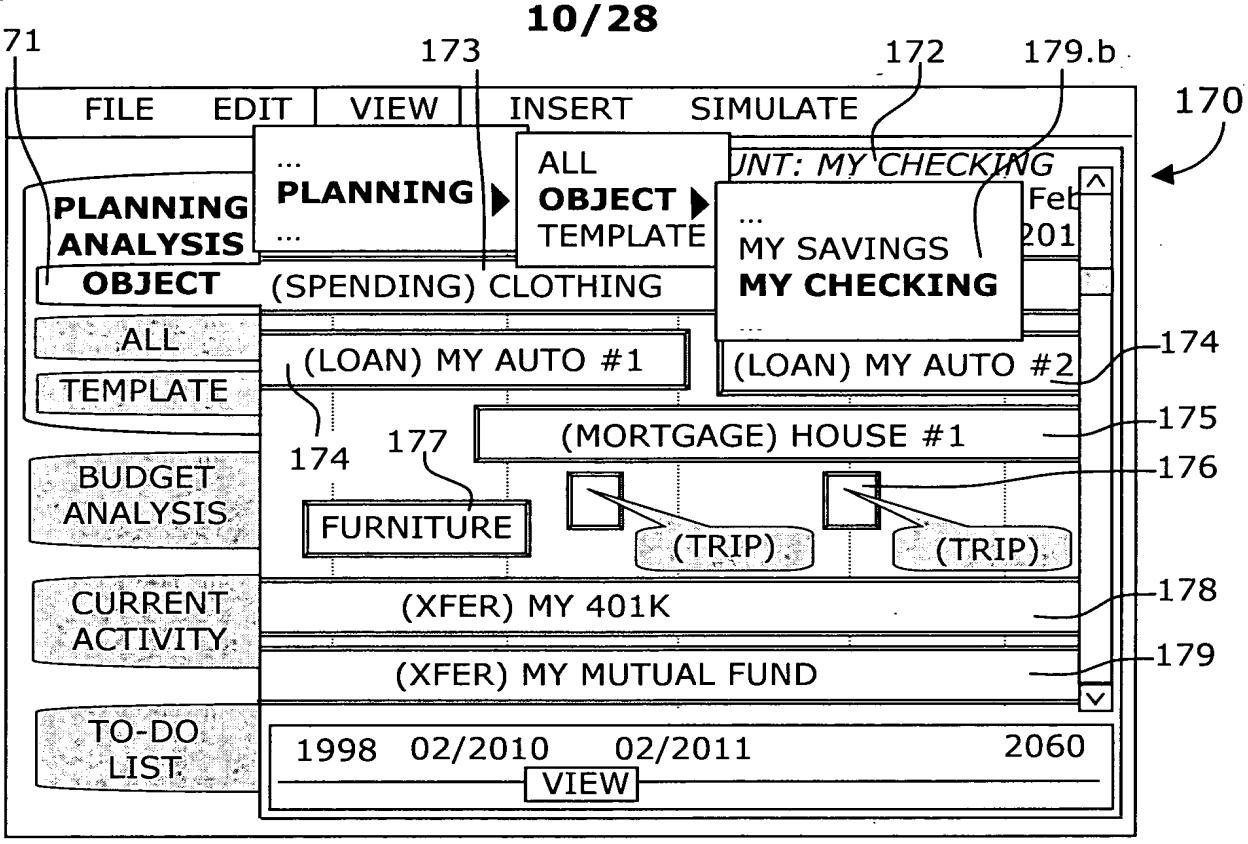

FIG. 17

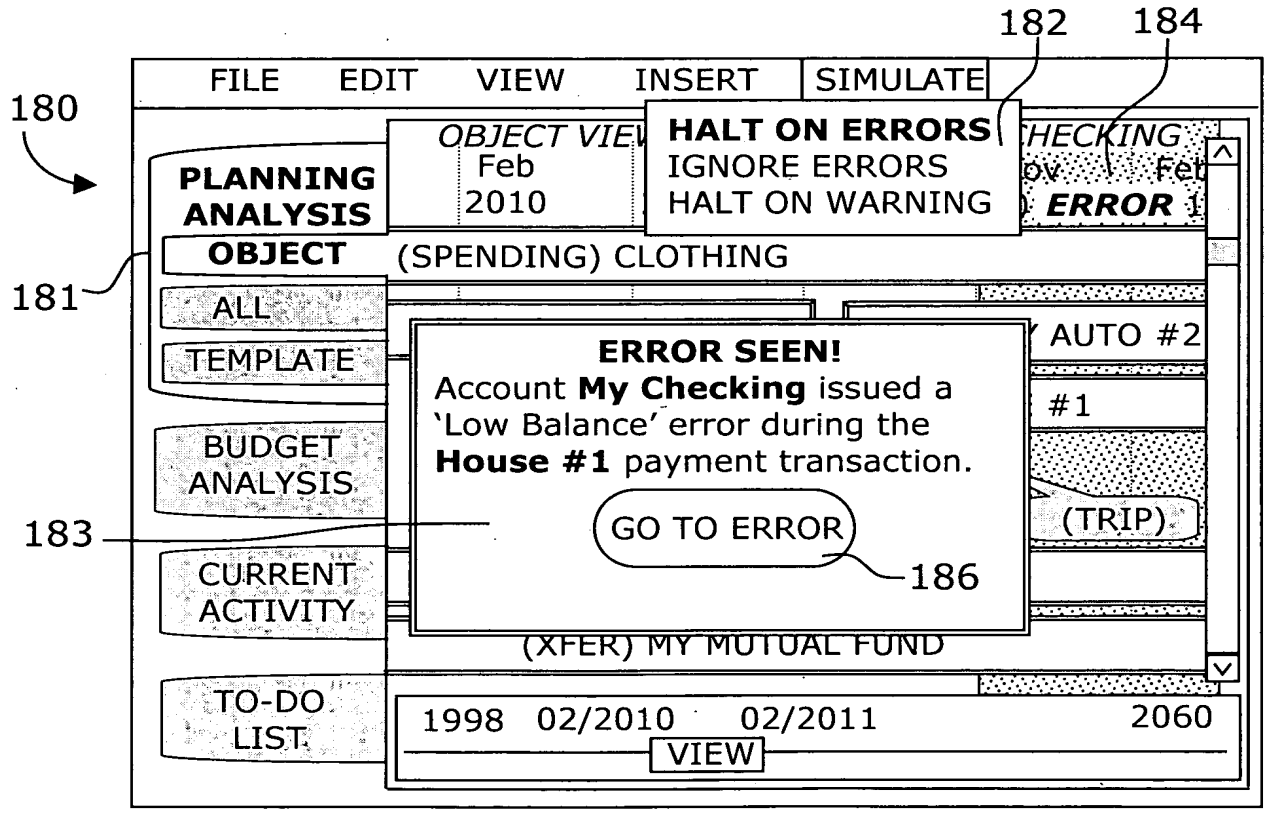

FIG. 18

OIPE
NOV 05 2004
PATENT & TRADEMARK OFFICE
JC156

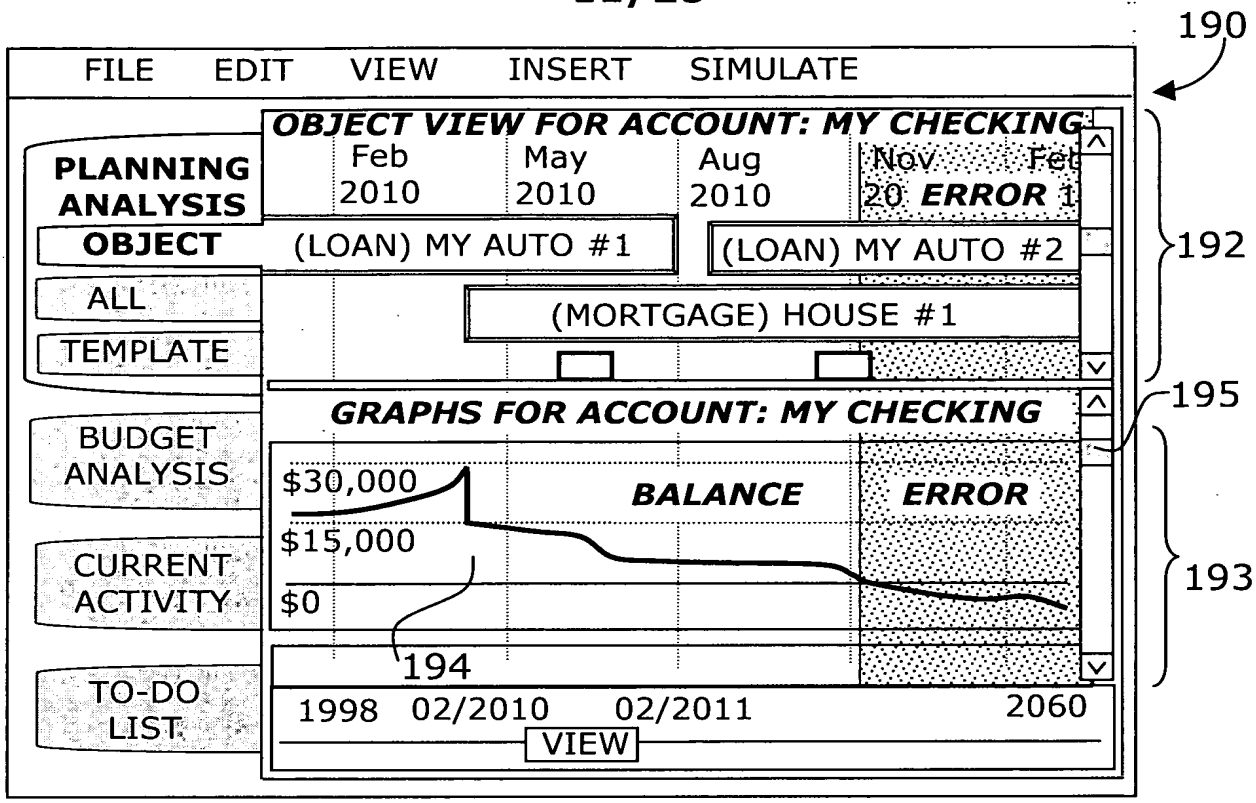

FIG. 19

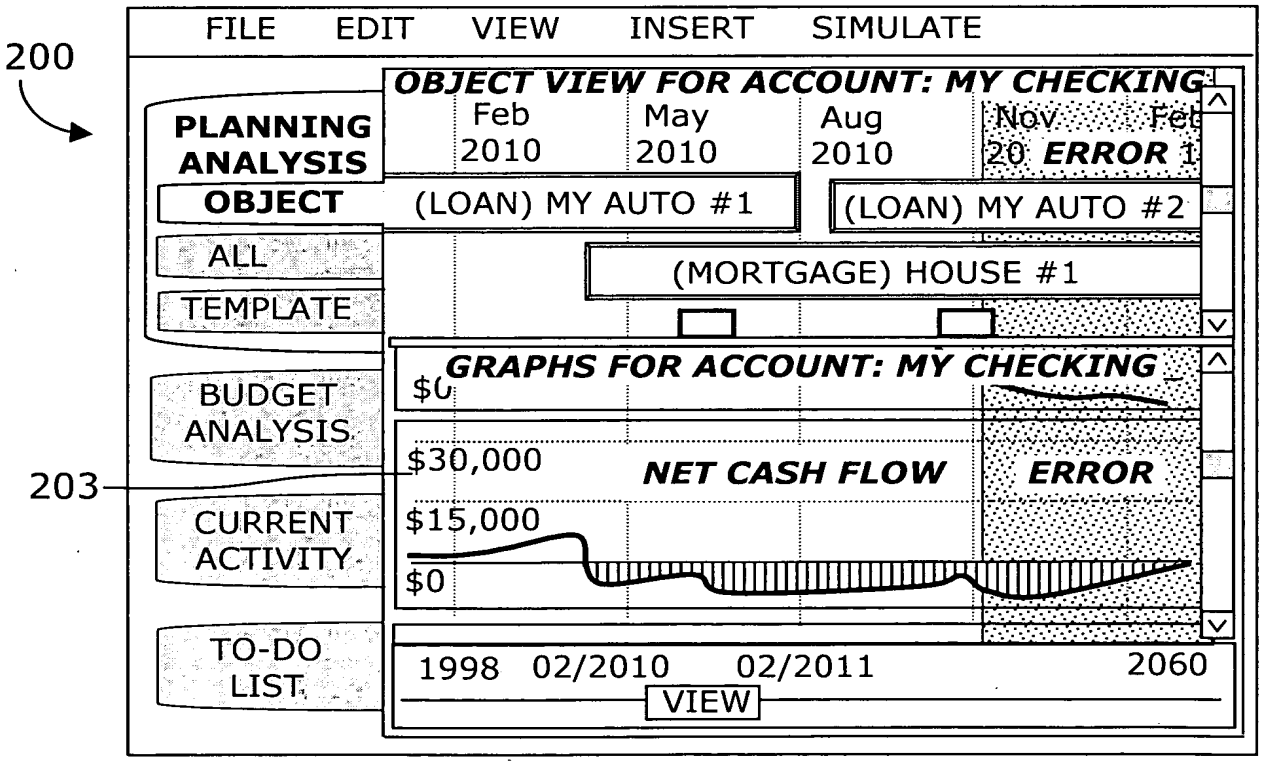

FIG. 20

OIPE
NOV 05 2004
PATENT & TRADEMARK OFFICE
JC156

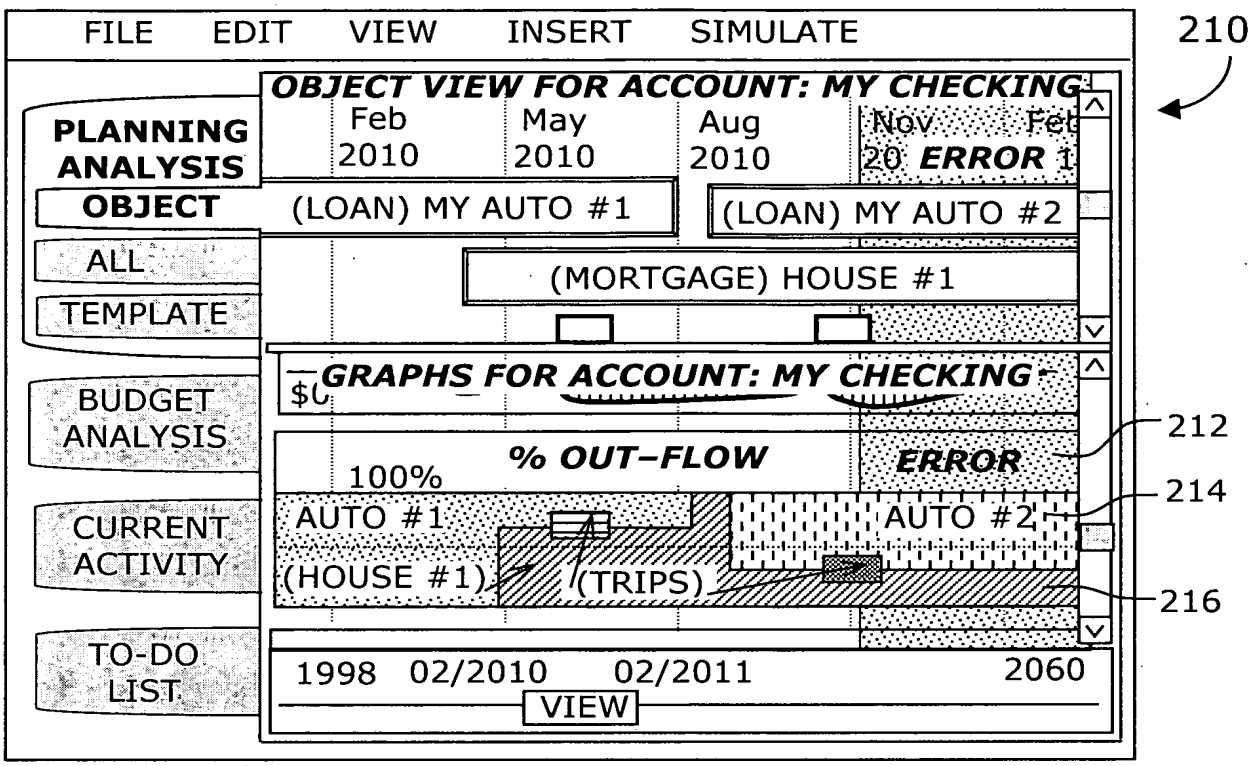

FIG. 21

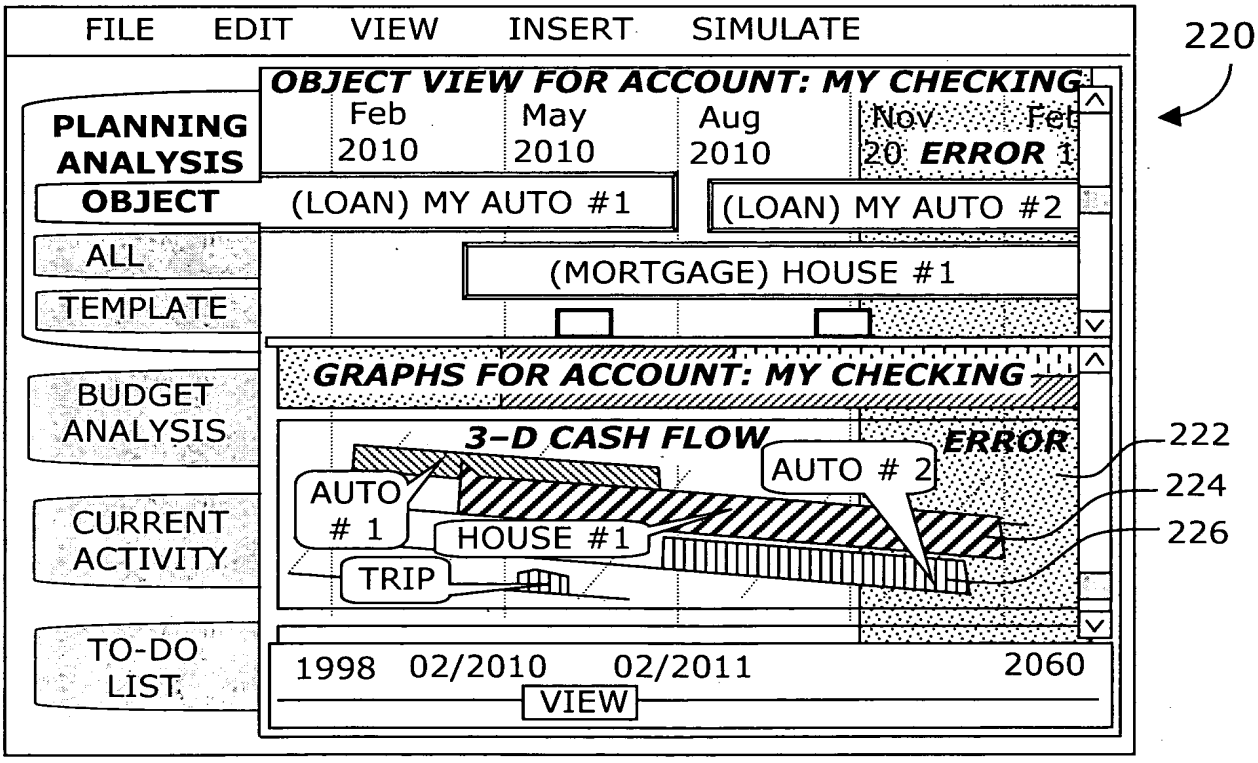

FIG. 22

OIPE
NOV 05 2004
JC156
PATENT & TRADEMARK OFFICE

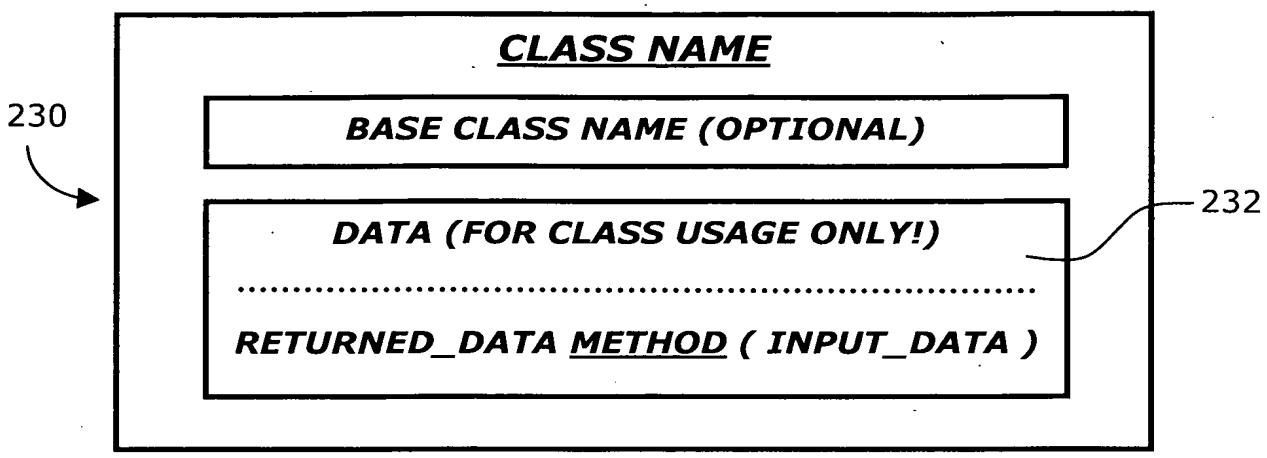

FIG. 23

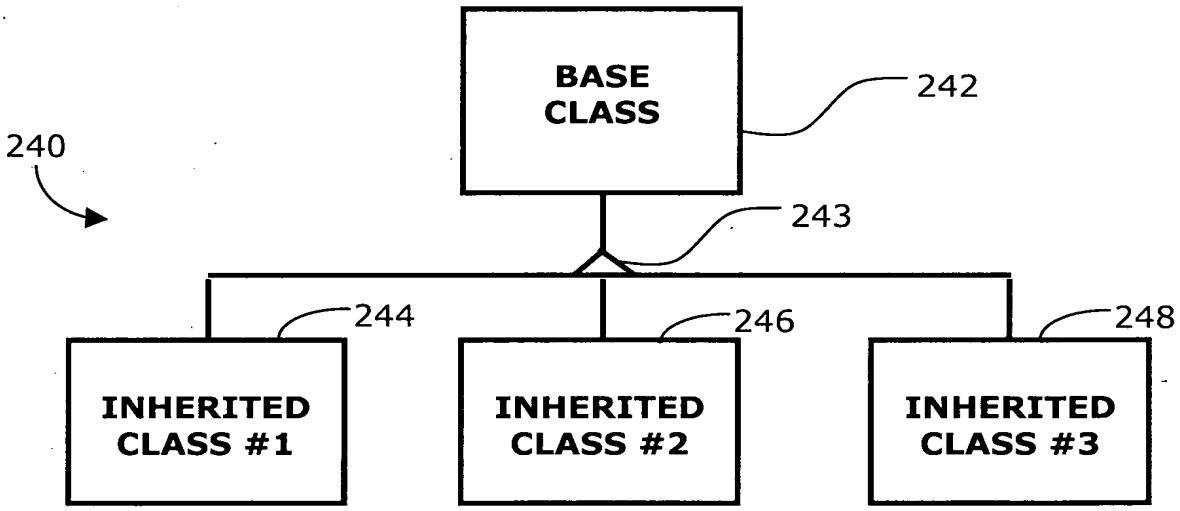

FIG. 24

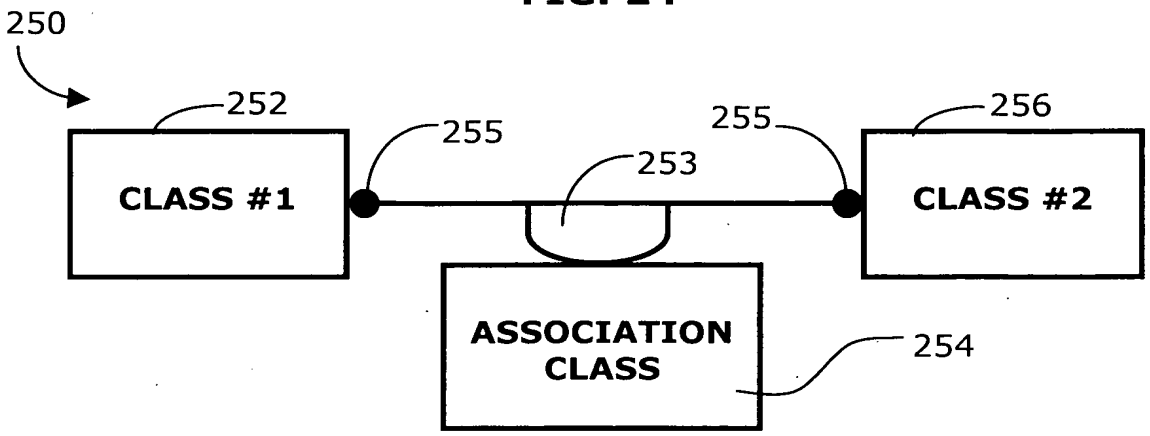

FIG. 25

OIPE
NOV 05 2004
PATENT & TRADEMARK OFFICE
JC156

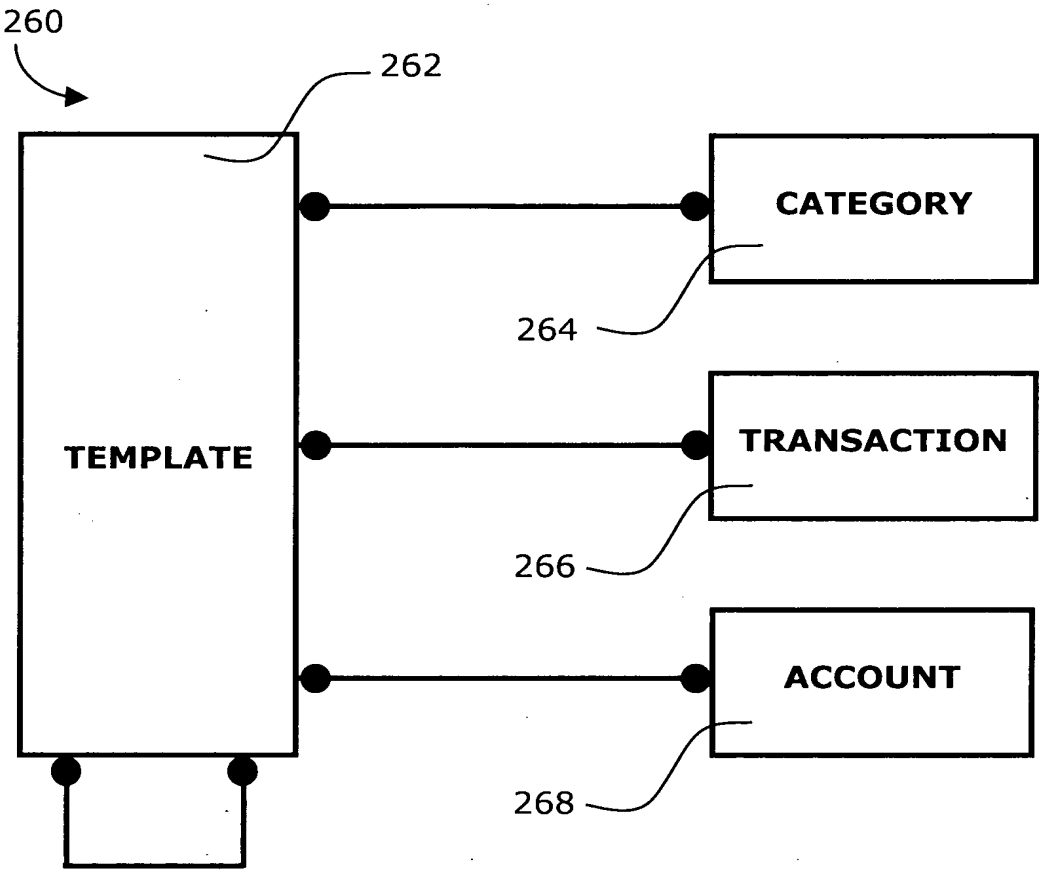

FIG. 26

PTO/SB/05 (08-03) Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.
UTILITY PATENT APPLICATION TRANSMITTAL
(Only for new nonprovisional applications under 37 CFR 1.53(b))
Attorney Docket No. 1000.1
First Named Inventor: Jeffrey Hammes
Express Mail Label No. EV 455 062 325 US
APPLICATION ELEMENTS
See MPEP chapter 600 concerning utility patent application contents.
1. Fee Transmittal Form (e.g., PTO/SB/17)
2. Applicant claims small entity status. See 37 CFR 1.27
3. Specification [Total Pages 28]
4. Drawing(s) (35 USC 113) [Total Sheets 28]
5. Oath or Declaration [Total Pages]
a. Newly executed (original or copy)
b. Copy from a prior application (37 CFR 1.63(d)) (for continuation/divisional with Box 16 complete)
i. DELETION OF INVENTOR(S)
Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
6. Application Data Sheet. See 37 CFR 1.76
7. CD-ROM or CD-R in duplicate, Large Table or Computer Program (Appendix)
8. Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary)
a. Computer Readable Form (CRF)
b. Specification Sequence Listing on CD-ROM or CD-R (2 copies); or paper
c. Statements verifying identity of above copies
ACCESSORY FEES
9. Assignment Papers (cover sheet & document(s))
10. 37 CFR 3.73(b) Statement (when there is an assignee)
11. English Translation Document (if applicable)
12. Information Disclosure Statement (IDS)/PTO-1449
13. Preliminary Amendment
14. Return Receipt Postcard (MPEP 503) (Should be specifically itemized)
15. Certified Copy of Priority Document(s) (if foreign priority is claimed)
16. Nonpublication Request under 35 U.S.C. 122(b)(2)(B)(i). Applicant must attach form PTO/SB/35 or its equivalent.
17. Other:
18. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information:
Continuation Divisional Continuation-in-part (CIP) of prior application No:
19. CORRESPONDENCE ADDRESS
Customer Number or Bar Code Label
Name (Print/Type) Jeffrey Hammes
Address 1000 Elwell Court, Suite 223
City Palo Alto State CA Zip Code 94303
Country USA Telephone 650-961-8300 Fax 650-961-8302
Date 11/5/2004
Signature
Burden Hour Statement: This collection of information is required by 37 CFR 1.53(b). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.
If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.
NOV 05 2004
JC156
TRADEMARK OFFICE

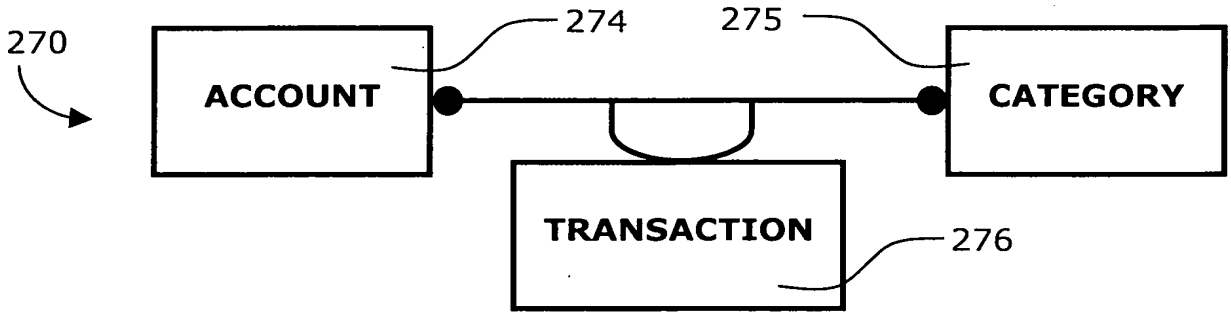

FIG. 27A

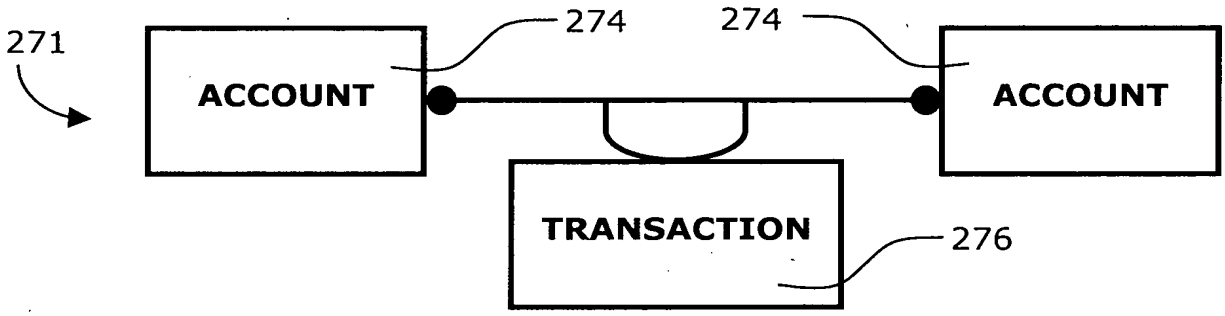

FIG. 27B

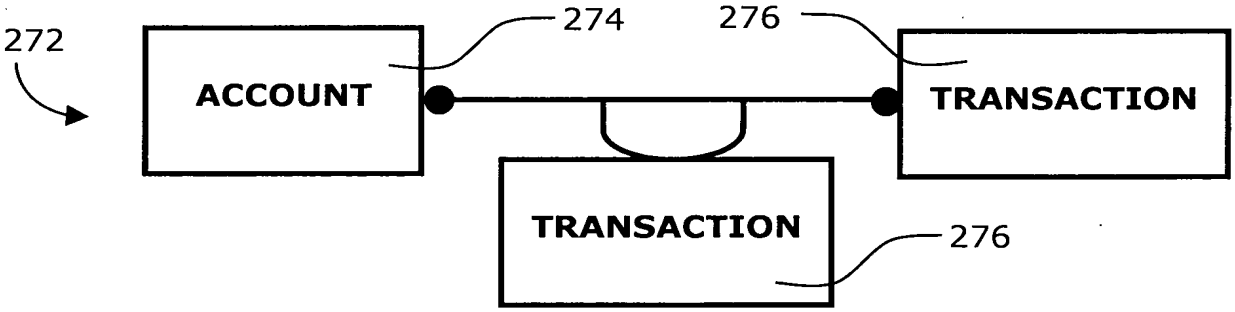

FIG. 27C

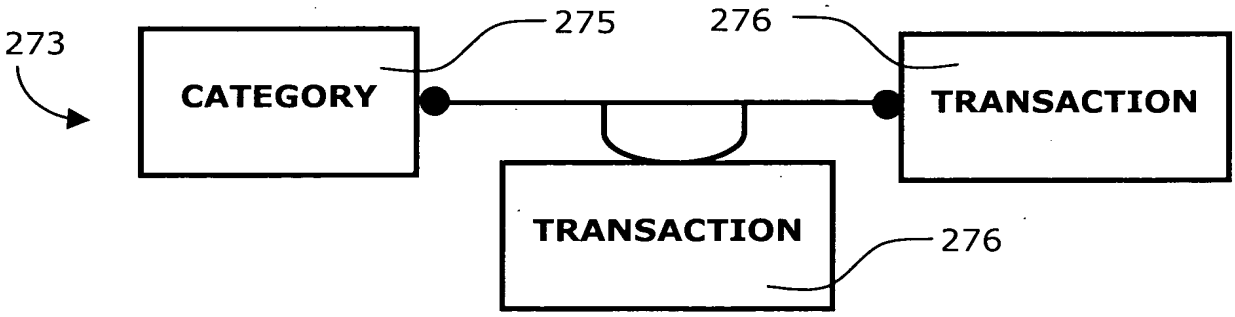

FIG. 27D

OIPE JC156
NOV 05 2004
PATENT & TRADEMARK OFFICE

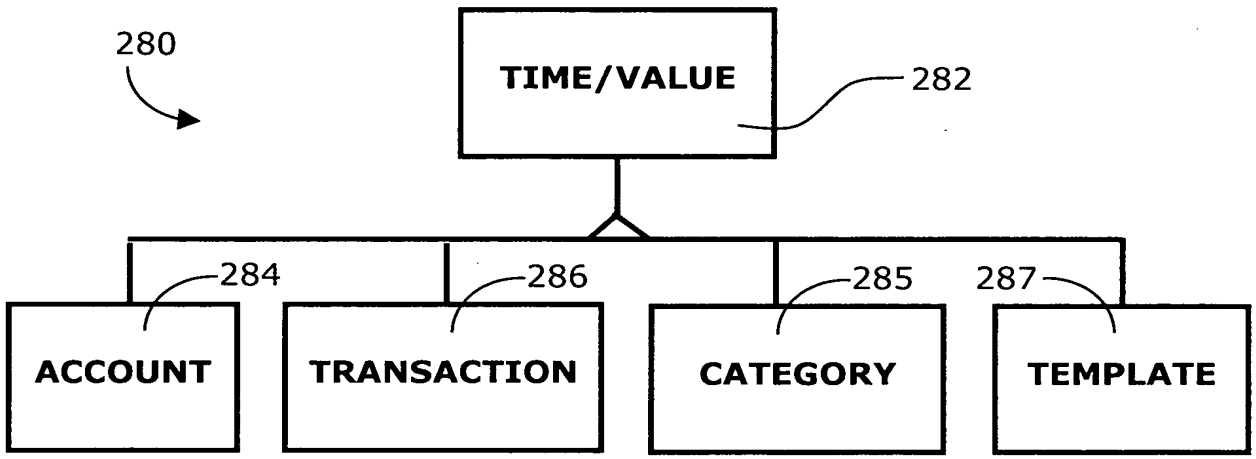

FIG. 28

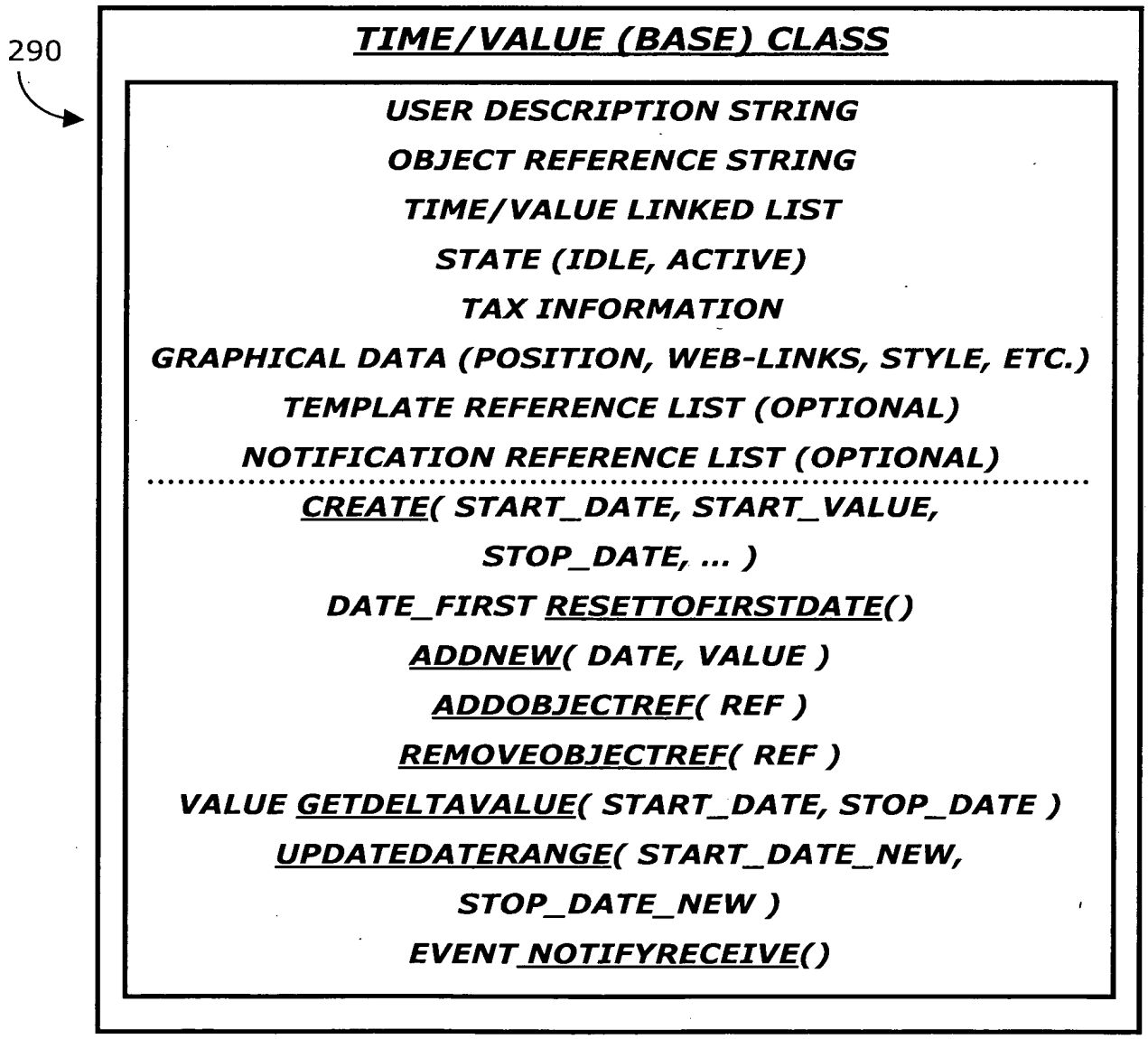

FIG. 29

OIPE JC156
NOV 05 2004
PATENT & TRADEMARK OFFICE

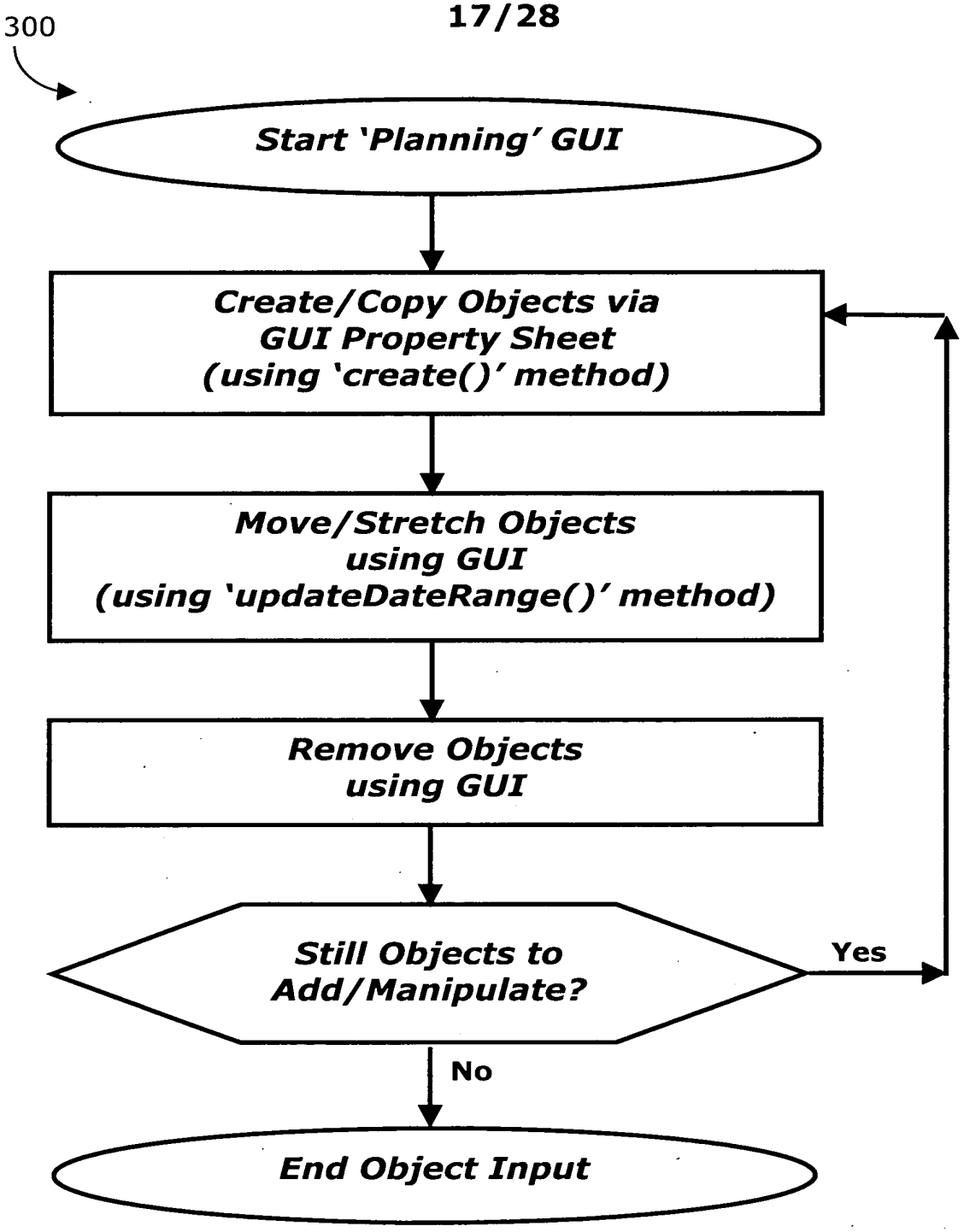

FIG. 30

OIPE JC156 NOV 05 2004 PATENT & TRADEMARK OFFICE

310

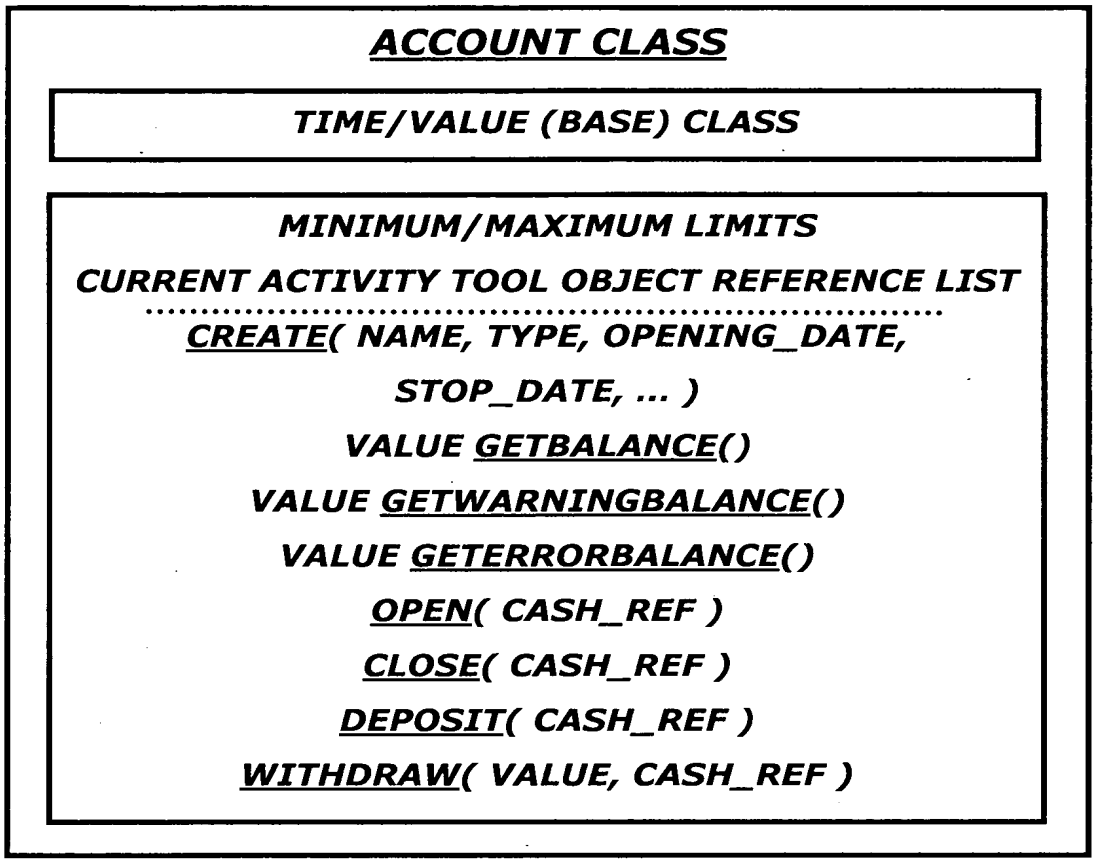

FIG. 31

320

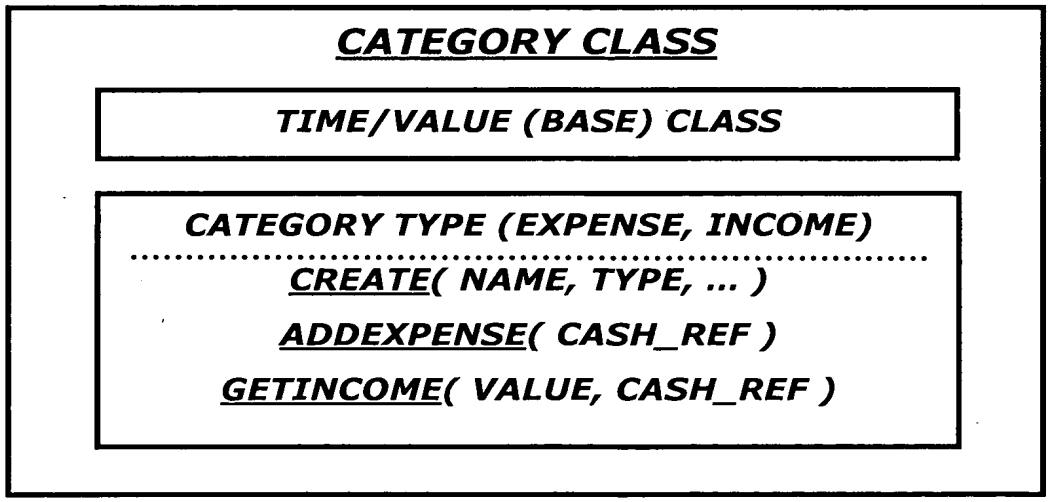

FIG. 32

OIPE JC156
NOV 05 2004
PATENT & TRADEMARK OFFICE

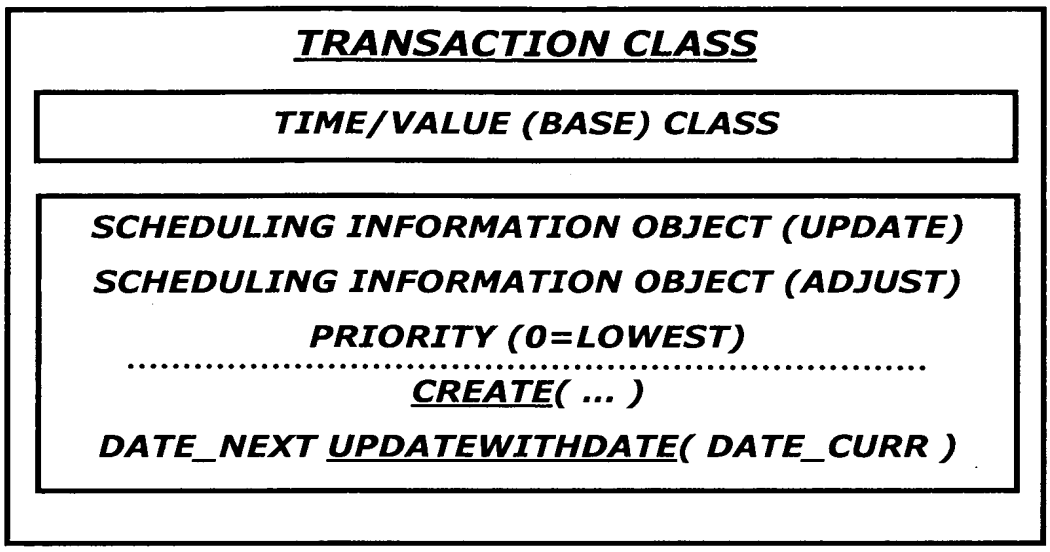

FIG. 33

OIPE JC156 NOV 05 2004 PATENT & TRADEMARK OFFICE

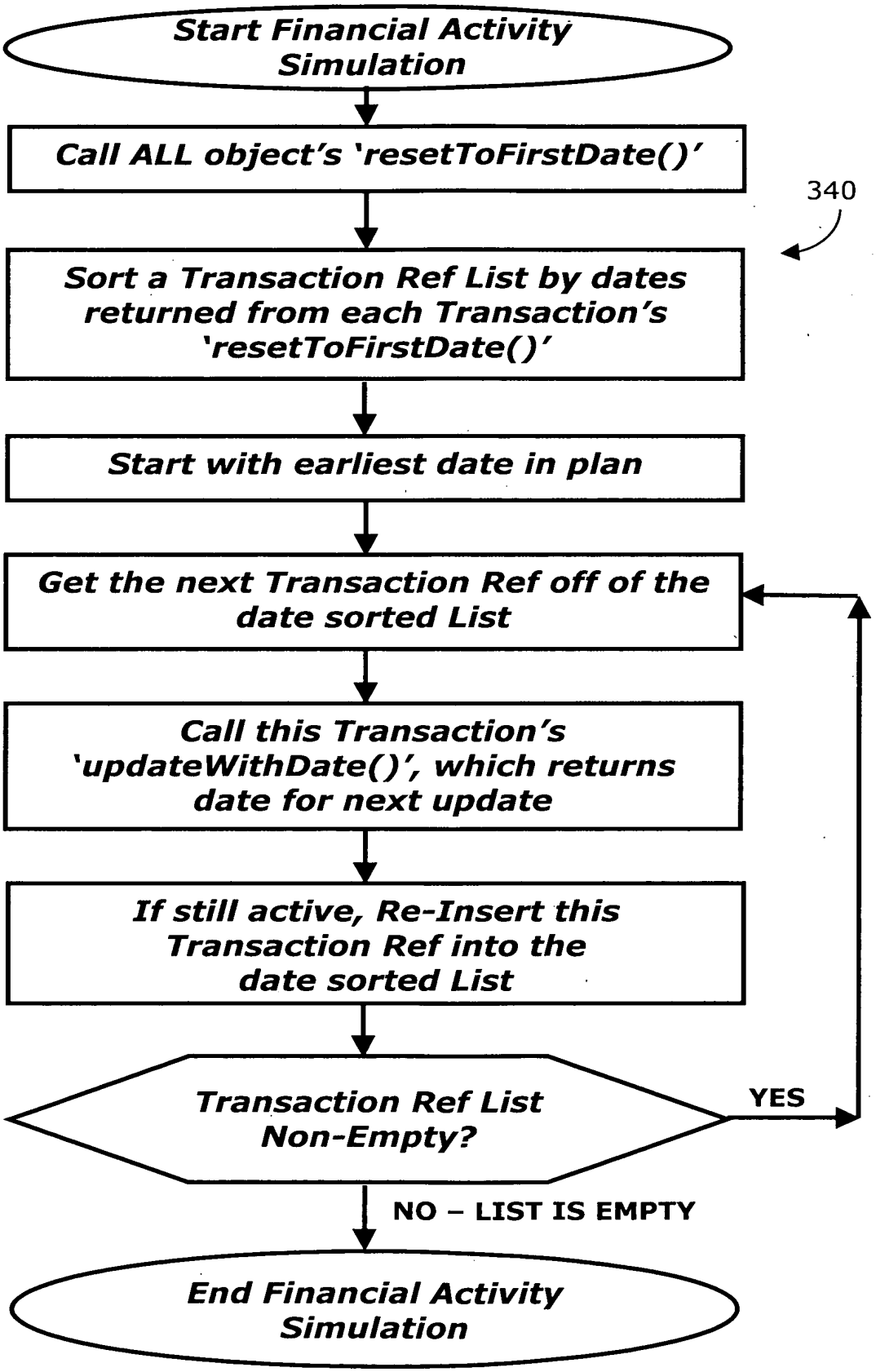

FIG. 34

OIPE
NOV 0 5 2004
PATENT & TRADEMARK OFFICE
JC156

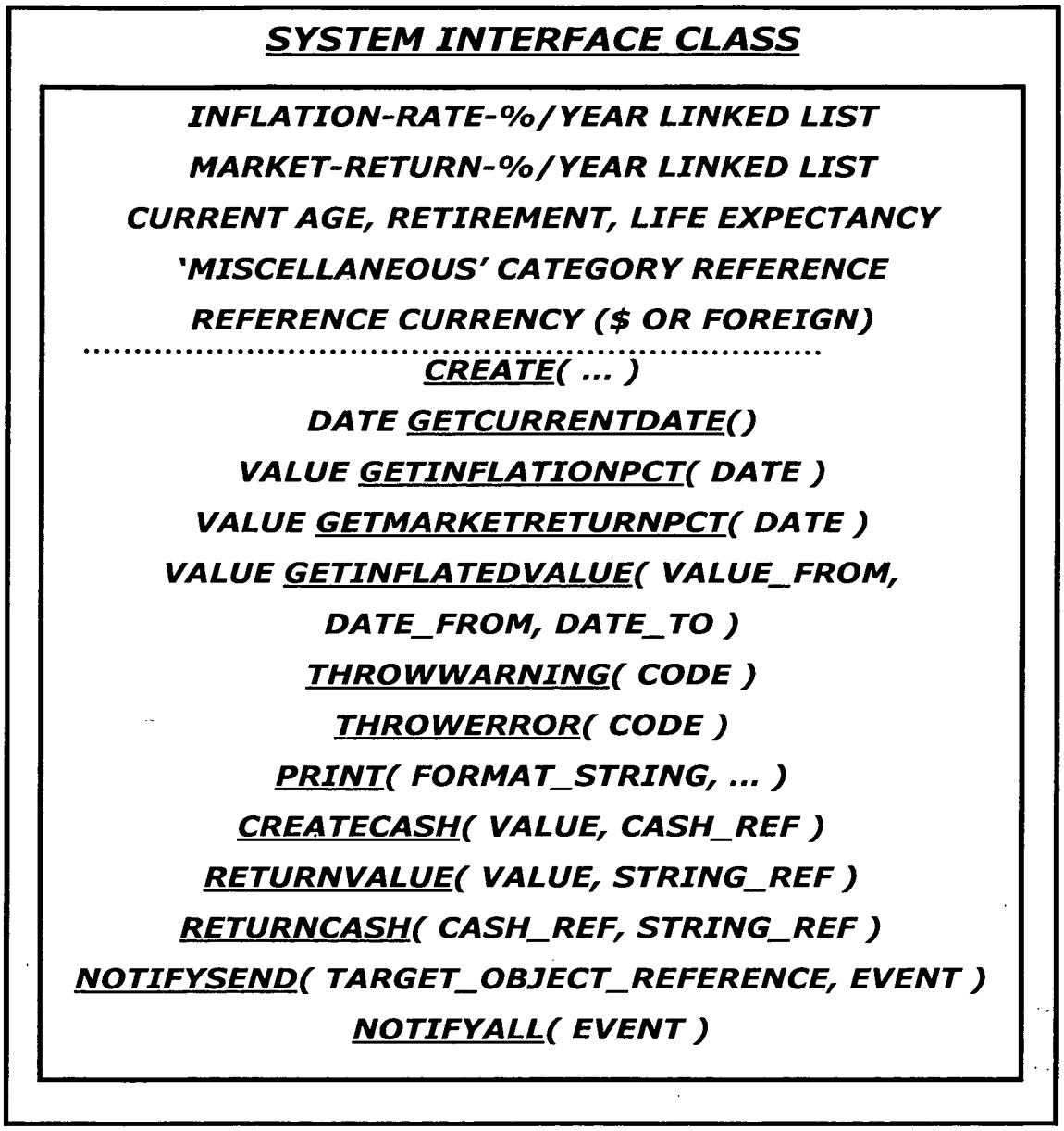

FIG. 35

OIPE JC156
NOV 05 2004
PATENT & TRADEMARK OFFICE

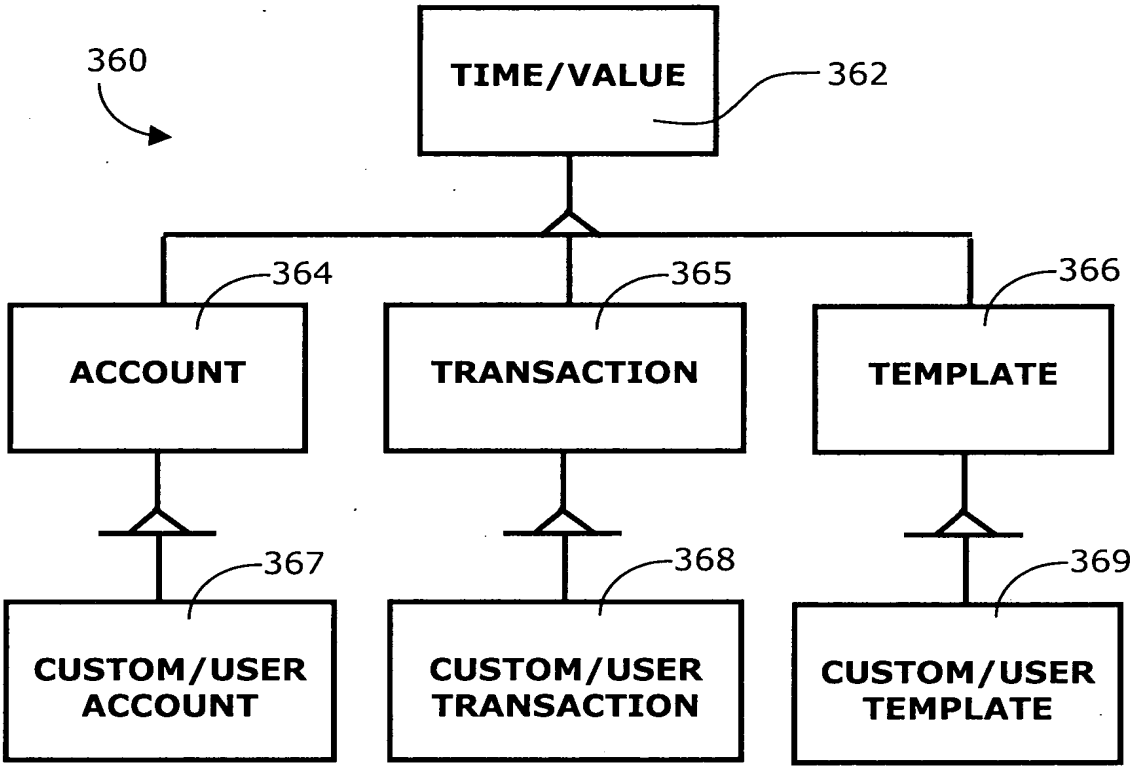

FIG. 36

OIPE
NOV 05 2004
JC156
PATENT & TRADEMARK OFFICE

370

SCHEDULING INFORMATION CLASS

FIRST DATE

LAST DATE

NEXT SCHEDULED DATE

SCHEDULING METHOD (DAILY, WEEKLY, MONTHLY, ETC.)

SCHEDULING FREQUENCY (EVERY TIME, EVERY OTHER TIME, EVERY 3RD, ETC.)

CREATE(...)

DATE RESETTOFIRSTDATE()

DATE GETNEXTDATE()

SETNEXTDATE(DATE)

DATE COMPUTENEXTDATE()

UPDATEDATERANGE(START_DATE_NEW, STOP_DATE_NEW)

FIG. 37

380

ACCOUNT-TO-ACCOUNT TRANSFER TRANSACTION CLASS

TRANSACTION (BASE) CLASS

'FROM' ACCOUNT OBJECT REFERENCE

'TO' ACCOUNT #2 OBJECT REFERENCE

TRANSFER AMOUNT VALUE

ADJUSTMENT PERCENTAGE

CREATE(...)

FIG. 38

OIPE JC156 NOV 05 2004 PATENT & TRADEMARK OFFICE

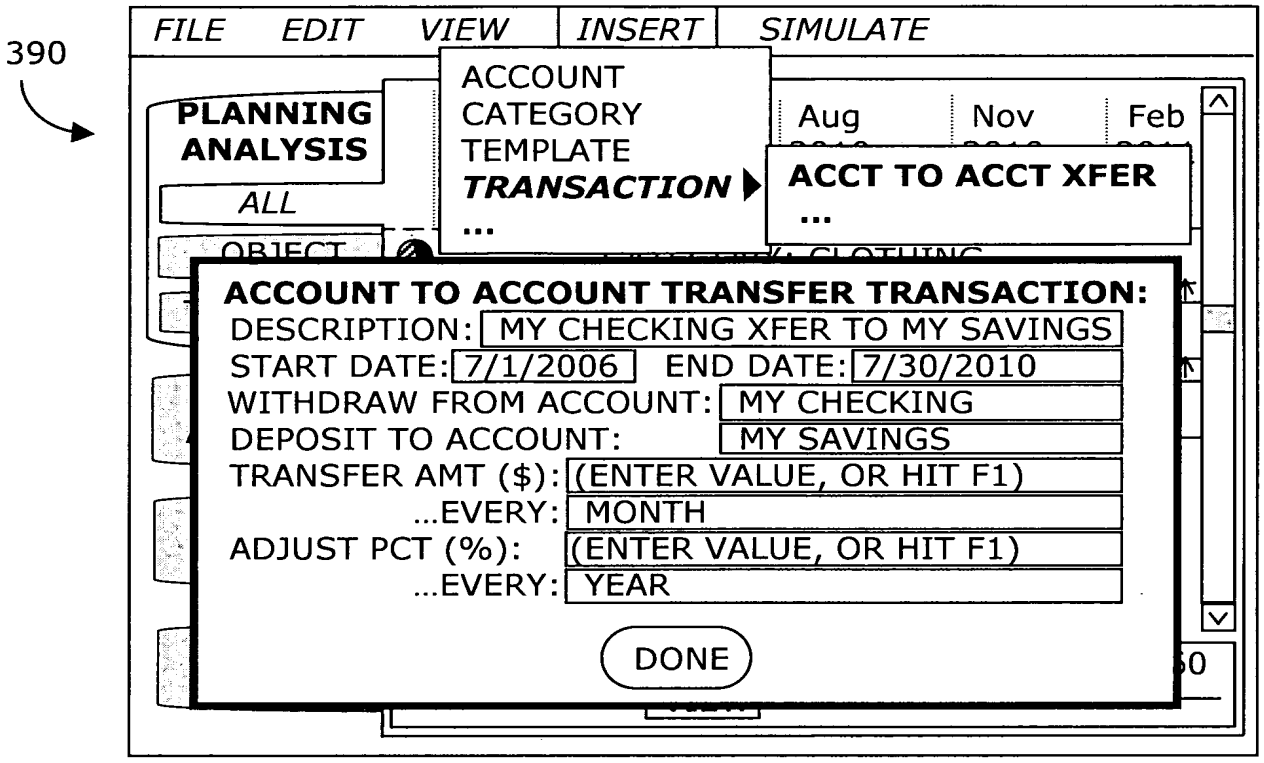

FIG. 39

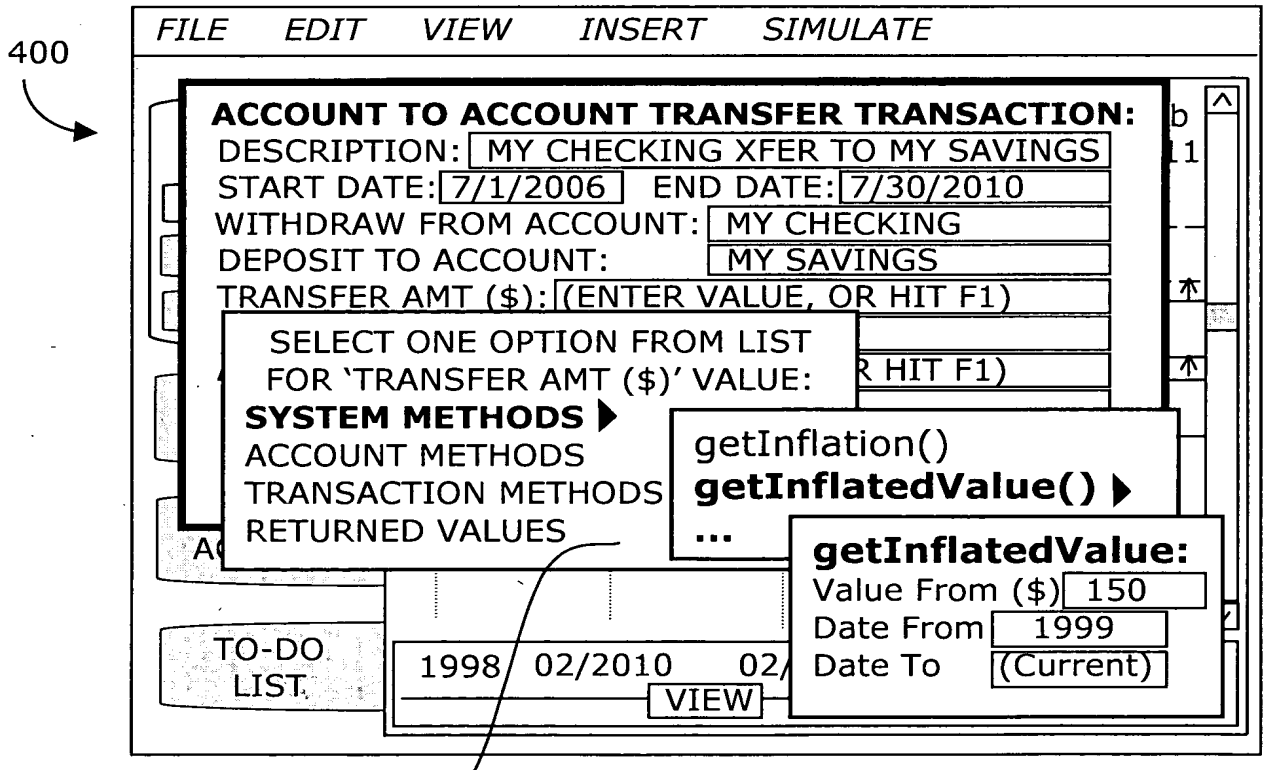

FIG. 40

OIPE JC156 NOV 05 2004 PATENT & TRADEMARK OFFICE

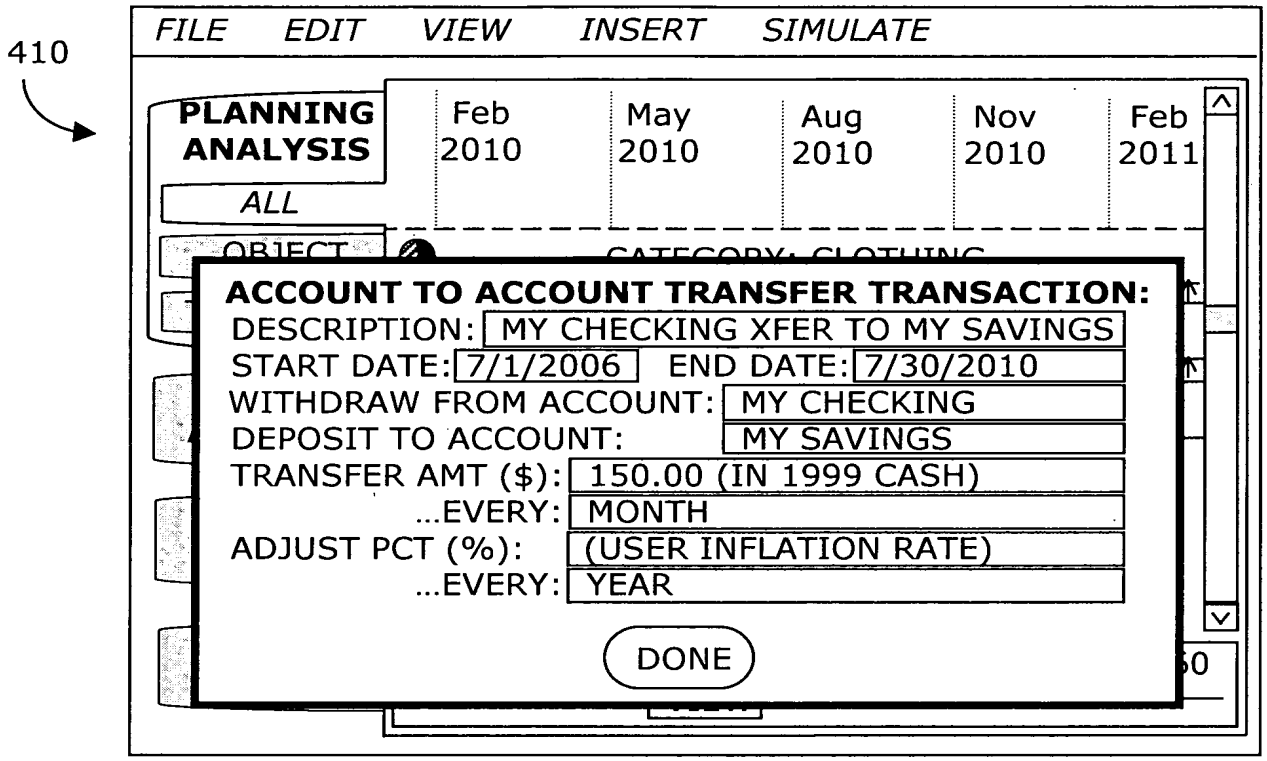

FIG. 41

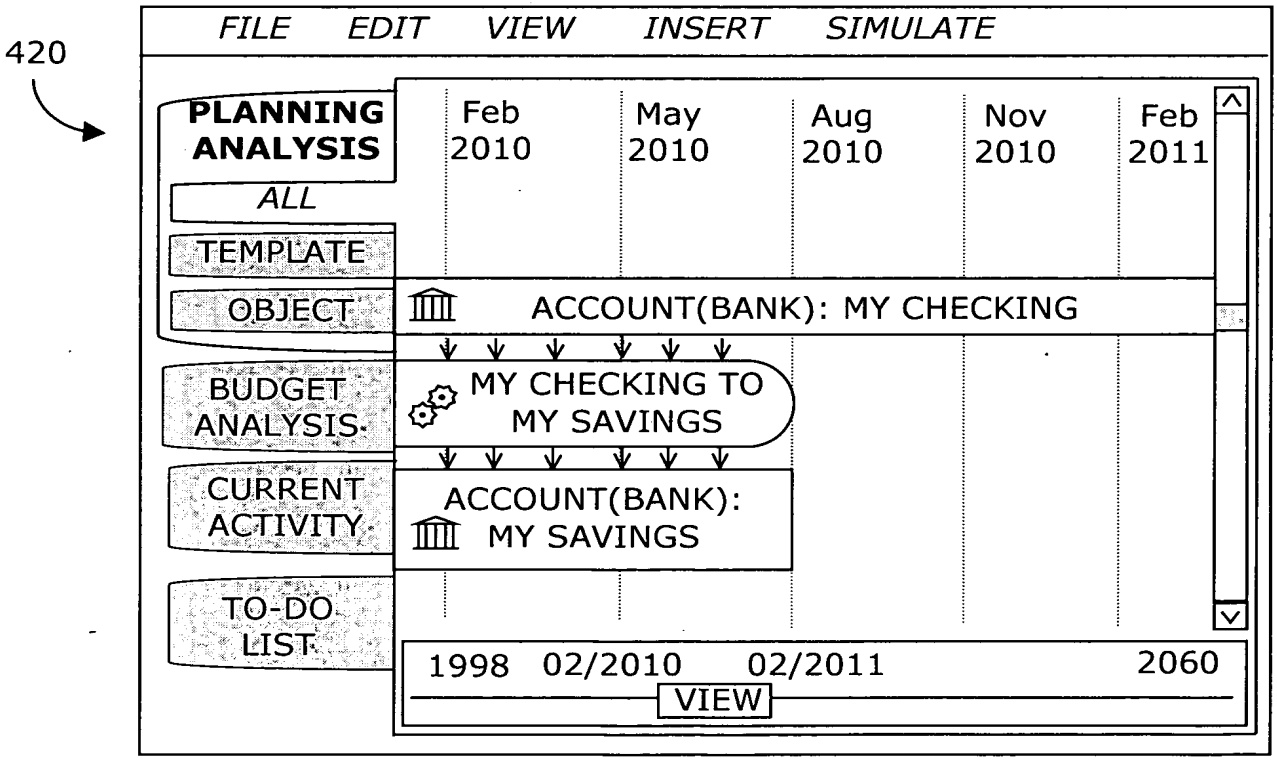

FIG. 42

OIPE JC156
NOV 05 2004
PATENT & TRADEMARK OFFICE

430

Transaction Class

```
class CTrans : public CtimeValue // A pure-virtual/abstract
                                 //    base class!
{
public:
   CTrans(
      const char *name, // Transaction's reference-name
      ... );                // More input parameters (not shown)
   virtual ~CTrans();

   virtual CDate updateWithDate( CDate date_curr ) = 0;
                 //PURE VIRTUAL!...Must inherit this class!

   virtual CDate resetToFirstDate();
                 // Inherited from Time/Value

   virtual void updateDateRange(
     CDate date_start, CDate date_stop );
                 // Inherited from Time/Value

protected:
   CScheduler  m_schUpdate; // Schedules next update date
   CScheduler  m_schAdjust;  // Schedules next adjust date
   priority_t    m_priority;     // Priority (0=lowest)

}; // END of 'CTrans' class
```

FIG. 43

OIPE JC156 NOV 05 2004 PATENT & TRADEMARK OFFICE

Account-to-Account Transfer Transaction Class

```
class CTrans_acctToAcct : public CTrans
{
public:
   CTrans_acctToAcct(
      const char *name,  // Transaction's reference-name
      ...     );          // More input parameters (not shown)
   virtual ~CTrans_acctToAcct();

   virtual CDate updateWithDate( CDate date_curr );

   virtual CDate resetToFirstDate();
   virtual void updateDateRange(
      CDate date_start, CDate date_stop );

protected:
   CAccount    *m_acctFrom; // Xfer 'From' this accnt
   CAccount    *m_acctTo;   // Xfer 'To' this accnt
   value_t  m_moneyToXfer;  // Money to transfer at
                            //   each update schedule
   value_t  m_adjustPct;    // % to adjust xfer amount

}; // END of 'CTrans_acctToAcct' class
```

440 →

FIG. 44

CASH CLASS

450 →

AMOUNT VALUE

CURRENCY TYPE (DOLLARS, POUNDS, ETC.)

CREATE()

ADDTOCASH(ADD_VALUE)

VALUE GETFROMCASH()

VALUE CURRENTAMOUNT()

FIG. 45

OIPE JC156
NOV 0 5 2004
PATENT & TRADEMARK OFFICE

Account-to-Account Transfer
Transaction Class Method Example

```
CDate CTrans_accntToAccnt::updateWithDate( CDate
date_curr ) {
   Cdate date_test =
     SYSINTF.getCurrentDate(); // Not used, just for demo
   if ( date_test == date_curr )
     SYSINTF.print( "Just a test...dates will be equal!" );
   // If simulated current date does NOT match our expected
   //    current date, leave (an invalid condition)
   if ( date_curr != m_schUpdate.getNextDate() ) {
     SYSINTF.throwError( ERR_UNEXPECTED_DATE );
     return(date_curr);// Ends simulation for this transaction!
   }

   // If current simulation date matches or exceeds
   //    our next adjustment date, adjust parameters
   if ( date_curr >= m_schAdjust.getNextDate() ) {
     m_moneyToXfer *= 1.0 + m_adjustPct / 100.0;
     m_schAdjust.computeNextDate(); }  // Set the next
           // adjustment date

   // CREATE 'cash' data type (simulated cash amount = 0)
   CCash cash_xfer;
   // WITHDRAW cash FROM account (simulate cash > 0)
   m_acctFrom->withdraw( m_moneyToXfer, cash_xfer );
   // DEPOSIT cash TO account (simulated cash = 0)
   m_acctTo->deposit( cash_xfer );

   // LOG this transfer amount to the Time/Value (base) class
   addNew( date_curr, m_moneyToXfer );
   // Return the date that we wish the Cash-Flow Simulator to
   //    call us with again
   return( m_schUpdate.computeNextDate() );
   // NOTE: Return from this method will cause 'cash_xfer'
   //    to be destroyed, calling 'cash' class' destructor
   //    method.  A NON-ZERO simulated cash amount in
   //    'cash_xfer' would cause a system warning!
} // END of 'CTrans_accntToAccnt::updatePerDate()'
```

460 →

FIG. 46